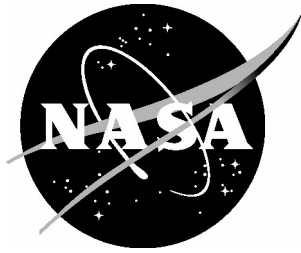


NASA/CR-2004-213021



# B-737 Linear Autoland Simulink Model

*Edward F. Hogge*  
*Lockheed Martin Corporation, Hampton, Virginia*

---

May 2004

## The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

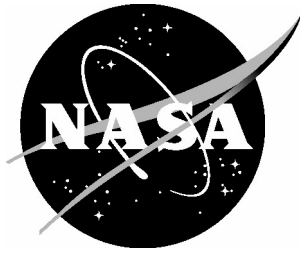
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at [\*http://www.sti.nasa.gov\*](http://www.sti.nasa.gov)
- E-mail your question via the Internet to [\*help@sti.nasa.gov\*](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:  
NASA STI Help Desk  
NASA Center for AeroSpace Information  
7121 Standard Drive  
Hanover, MD 21076-1320

NASA/CR-2004-213021



# B-737 Linear Autoland Simulink Model

*Edward F. Hogge*  
*Lockheed Martin Corporation, Hampton, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NAS1-00135

May 2004

## Acknowledgments

I would like to thank Dr. Alan White for his assistance in the description of the Dryden gust model mathematics and the random number generator repair. I would like to thank Arlene Guenther of Unisys for help with understanding and adapting the original FORTRAN Airlabs model and the document that this is adapted from. Also, thanks to Dr. L. Keith Barker for his help with reference frames and general mathematical analysis suggestions.

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.
---

Available from:

NASA Center for AeroSpace Information (CASI)  
7121 Standard Drive  
Hanover, MD 21076-1320  
(301) 621-0390

National Technical Information Service (NTIS)  
5285 Port Royal Road  
Springfield, VA 22161-2171  
(703) 605-6000

# Contents

<b>CONTENTS .....</b>	<b>1</b>
<b>INTRODUCTION .....</b>	<b>3</b>
HISTORY OF LINEAR AUTOLAND .....	3
CONTROL SYSTEM DESCRIPTION .....	4
DEVELOPMENT EXPERIENCE .....	4
<b>MODEL ASSUMPTIONS .....</b>	<b>5</b>
CHECK CASES .....	5
FLIGHT CASE 1 .....	6
FLIGHT CASE 2 .....	7
<b>MODEL DESCRIPTION.....</b>	<b>7</b>
MAJOR FUNCTIONAL BLOCKS .....	7
<b>NEWTON'S LAWS OF MOTION AND THE MODEL .....</b>	<b>10</b>
FORCE AND TORQUE IN THE INERTIAL REFERENCE FRAME .....	11
BODY REFERENCE FRAME FOR SIMPLIFIED TORQUE CALCULATION .....	11
FORCE AND TORQUE IN THE NON INERTIAL REFERENCE FRAME OF THE AIRCRAFT .....	14
<b>REFERENCE FRAMES.....</b>	<b>15</b>
BODY REFERENCE FRAME .....	16
INERTIAL REFERENCE FRAME.....	18
ORIENTATION USING EULER ANGLES.....	20
CARTESIAN VELOCITY IN INERTIAL FRAME .....	21
ANGULAR VELOCITY IN INERTIAL FRAME.....	24
AUTOPILOT RUNWAY FRAME .....	25
<b>STATE SPACE MODEL .....</b>	<b>26</b>
STATE VECTOR DESCRIPTION .....	28
<i>FORTTRAN Common for State Vector.....</i>	28
CONTROL INPUT VECTOR DESCRIPTION .....	29
<i>FORTTRAN Common for Control Input Vector.....</i>	30
INERTIAL STATE VECTOR .....	30
<i>Simulink Implementation of the State Space Model .....</i>	31
<b>EQUATIONS OF MOTION.....</b>	<b>33</b>
LOCALIZER AND GLIDE SLOPE ERROR.....	34
<b>DRYDEN GUST MODEL .....</b>	<b>37</b>
DRYDEN GUST MODEL NOTATION .....	40
LAPLACE TRANSFORM TRANSFER FUNCTION DERIVATION .....	42
SPECTRAL FACTORIZATION .....	45
GAUSSIAN BAND LIMITED WHITE NOISE.....	49
Z-TRANSFORM TRANSFER FUNCTION DERIVATION .....	53
LAG FILTER CASE .....	63
LAG LEAD FILTER CASE .....	72
RANDOM NUMBER GENERATOR.....	76
<b>AUTOPILOT .....</b>	<b>81</b>
CONTROL CONCEPTS .....	81

AUTOTHROTTLE.....	85
LATERAL AUTOPILOT .....	87
<i>Lateral Autopilot Simulink Diagram</i> .....	88
LONGITUDINAL AUTOPILOT.....	89
<i>Longitudinal Autopilot Simulink Diagram</i> .....	90
<i>Vertical Path Control</i> .....	91
<i>Autoland</i> .....	92
<i>Flight Modes</i> .....	93
<b>ENGINE MODEL .....</b>	<b>95</b>
<i>International Standard Atmosphere (ISA)</i> .....	97
ENGINE PRESSURE RATIO .....	98
COMPRESSOR BLEED VALVE OPERATION .....	101
<b>LONGITUDINAL CONTROL ACTUATORS AND SERVOS .....</b>	<b>103</b>
ELEVATOR .....	103
STABILATOR .....	106
<b>LATERAL CONTROL ACTUATORS .....</b>	<b>108</b>
AILERON .....	108
SPOILER .....	110
RUDDER.....	111
<b>APPROACH TO CONVERSION .....</b>	<b>113</b>
<i>Summary of FORTRAN to Simulink Porting Process</i> .....	114
<b>CONCLUSION .....</b>	<b>116</b>
<b>REFERENCES .....</b>	<b>117</b>

## Abstract

*The Linear Autoland Simulink model was created to be a modular test environment for testing of control system components in commercial aircraft. The input variables, physical laws, and reference frames used are summarized. The state space theory underlying the model is surveyed and the location of the control actuators is described. The equations used to enable the Dryden gust model to simulate winds and gusts are derived. A description of the pseudo-random number generation method used in the wind gust model is included. The longitudinal autopilot, lateral autopilot, automatic throttle autopilot, engine model and automatic trim devices are considered as subsystems. The experience in converting the Airlab FORTRAN aircraft control system simulation to a graphical simulation tool (Matlab/Simulink) is described.*

## Introduction

The Linear Autoland Simulink model was created to provide a modular simulation environment for testing of control system components in the Systems and Airframe Failure Emulation Testing and Integration (SAFETI) Laboratory. The Langley B-737-100 research aircraft delivered in 1973 was the object of the model. The B-737-100 was the first production model of the B-737 series, and was a short-range two-engine transport that carried 85-99 passengers. More than 2700 aircraft of the B-737 family have been delivered and have made more than 62 million flights. The laboratory simulation was derived from the FORTRAN source code of a simulation developed under contract by the Sperry Corporation for the Advanced Transport Operating System (ATOPS) project office of Langley Research Center in 1985. The Sperry simulation was a linear version of the ATOPS B-737 non-linear simulation for use in Airlab, a Langley control systems laboratory. The FORTRAN code was subsequently modified to get it to operate correctly on a Sun SPARCstation host. The Dryden gust model scaling coefficients were corrected to agree with reference documentation and its random number generator was replaced to correct a problem with periodicity.

The simulation consists of an aerodynamics state space model of the B-737 airframe and actuators. The derivative equations are integrated to get parameters in the aircraft frame of reference and parameters to locate the aircraft in an earth-fixed frame of reference. A runway frame of reference is used to align the glide slope and track angle during landing. The autopilot uses the glide slope and track angle reference orientation to adjust the aircraft's course and attitude. Wind gusts are simulated using the Dryden gust model. The wind gusts and autopilot control inputs act on the airframe plant dynamics model. Some simplifying assumptions had been made previously in the ATOPS B-737 FORTRAN simulation to fit within limited computing power, and to make the simulation simpler.

### History of Linear Autoland

The simulation of Transport Systems Research Vehicle (TSRV) was the source of the linear Autoland simulation. It was converted by Sperry from FORTRAN 66 on the Control Data Corporation (CDC) Network Operating System (NOS) to FORTRAN 66 on the Digital Equipment Corporation (DEC) VAX 750 for use by the Airlab project. The linear version was chosen for several reasons. It was a way of preventing release of Boeing

proprietary aerodynamic data present in the full non-linear version. The target host VAX 750 was not fast enough to run the full non-linear equations. Also, too much programming labor would have been required to convert the non-linear aerodynamic data in CDC packed binary format to VAX format. The linear VAX version was good enough to satisfy the research requirements of Airlab.

Lockheed Martin was contracted to port the Airlab linear Autoland version from the VAX environment to the Sun Microsystems Unix workstation FORTRAN 77 environment for the SAFETI Laboratory. Subroutine calling sequences and parameter lists were aligned to correct repeatability problems. The Dryden gust model was repaired by changing filter coefficients to agree with published literature and by replacing a defective random number generator. A gust response of an order of magnitude larger was the result in the reference flight case 1, and was in agreement with the expectations of a NASA control engineer familiar with the Boeing 737 response to control inputs.

### **Control system description**

The control system consists of three major sections, the lateral autopilot, the longitudinal autopilot, and the automatic throttle. The lateral autopilot uses the spoilers, ailerons, and rudder to control the flight path left to right according to a heading set point. The longitudinal autopilot uses the elevator and stabilator to control the glide slope angle up and down according to a glide slope set point. The automatic throttle uses a target air speed to adjust the engine throttle in order to maintain a constant air speed. During the automatic landing phase of the flight, the lateral and longitudinal set points are replaced by track and glide slope error derived from radio transmitters at the destination runway. The error in heading is provided by the difference between the orientation of the aircraft path and a line to the localizer radio transmitter. The error in glide slope is provided by the difference between the flight path and a line to the glide slope transmitter. The automatic throttle maintains a constant airspeed. This situation continues until the final phase of the landing.

### **Development experience**

Lockheed Martin was contracted to port the FORTRAN Linear Autoland model to Matlab/Simulink. Matlab and Simulink are software tools for engineering analysis produced by The MathWorks Inc. They are commonly used by universities and engineering companies for control system simulation and development. The FORTRAN model was converted to Simulink according to a black-box procedure. The unchanged FORTRAN code was called from Simulink using Matlab dynamically linked subroutine (FORTRAN MEX) blocks. The FORTRAN modules used were those which modeled the aerodynamics and equations of motion. An open-loop configuration to simulate a straight, level flight condition was tested and explored. Once that configuration was operating repeatably, time histories of control parameters were substituted for the straight, level flight condition as input to the model. Corrections were made until the model parameters roughly matched the reference FORTRAN simulation runs. Next, subsystems of the FORTRAN model were replaced step-wise by Simulink blocks. When the Simulink blocks had completely replaced the FORTRAN subroutines, work began on the autopilot. A Simulink version of each autopilot subsystem was substituted for the corresponding time history input. This closed the loop in the simulation. Sources of error were isolated by detailed comparison of model time histories with the FORTRAN reference runs.



## Model Assumptions

The Sperry *Linear 737 Autoland Simulation for Airlabs* [Ref. 9] made simplifying assumptions to protect Boeing proprietary aerodynamic data, and to make analysis easier. Because the Airlab version was constrained by limited computer power, little attention was given to conversion of the Dryden gust model from the CDC NOS FORTRAN 66 implementation to the Airlab version because Airlab had few requirements for testing gust response. Here is a summary of the assumptions present in the Airlab's FORTRAN simulation and carried over to the B-737 Linear Autoland Simulink model.

- Linearized State Space Model
- No Boeing Proprietary Data
- Similar Dynamics and Control (A and B) Matrices for Flight Cases 1 and 2
- One Engine Instead of Two
- (The thrust is half that of the actual airplane. The state space model works as if the other engine were there. The throttle setting is the same as the actual airplane.)
- Flight Ends After Flare But Before Roll Out
- Constant Flaps, Set At 40 deg
- Center of Mass (Gravity) Set At 0.2 Fraction of Length of Fuselage or At 20% Mean Aerodynamic Chord (MAC)
- Gross Weight 85000 lbs
- Length 94 ft
- Wingspan 93 ft
- Tail Height 37 ft
- Langley AFB Runway 8 Approached From the West

### Check Cases

In order to prevent the introduction of errors into the simulation during the software conversion, time histories of important parameters were generated. The only information available from the original CDC NOS computer runs was strip chart time histories of flight case 2 engineering parameters. These strip charts were used to diagnose problems introduced into the FORTRAN simulation at the time of the conversion to run on the Airlab VAX. The software modifications necessary to run on the Sun were checked also. Once a valid and repeatable FORTRAN simulation was achieved on the Sun equipment, reference files of parameter time histories were saved. The FORTRAN source code configuration was maintained using the UNIX SCCS commands. The version described in

this document is g134v1.

Figures 1 and 2 depict the path of the simulated aircraft for flight cases 1 and 2, respectively, and describe the flight conditions.

### Flight Case 1

- Straight in approach, below glide slope along extended center line
- Capture glide slope from below
- Wind from Northeast, 20 knots

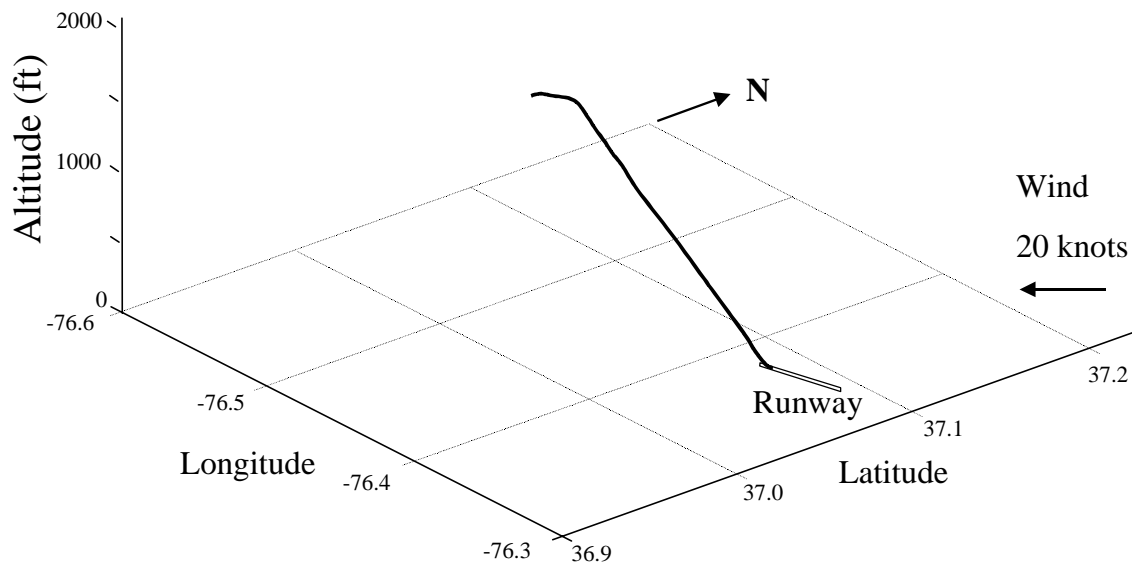


Figure 1. Path of simulated aircraft for flight case 1.

There is a sharp pitch change at the beginning of flight case 1. It is a good test to find stability problems with the longitudinal control system. It does not exercise the lateral control system at all.

## Flight Case 2

- Turning from base to final approach. Bank to left to capture glide slope, 130 degs. localizer capture followed by glide slope capture
- No wind

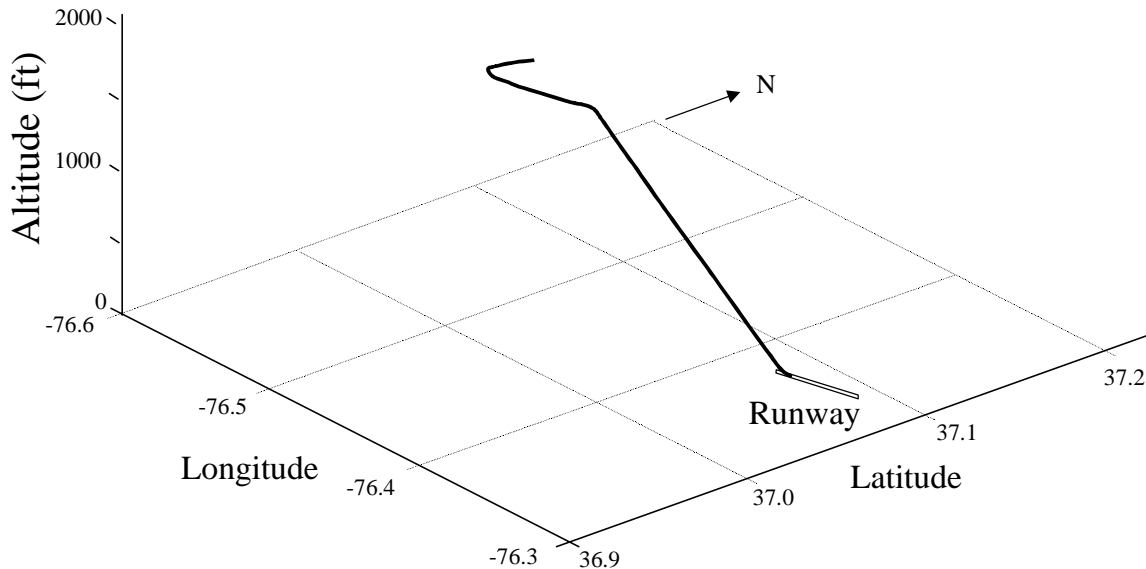


Figure 2. Path of simulated aircraft for flight case 2.

The bank to the left tests if the centrifugal and Coriolis forces are accounted for correctly. The lateral control system is exercised. Interaction of the ailerons with the spoilers is tested as well.

## Model Description

### Major Functional Blocks

The FORTRAN linear Autoland simulation consists of a series of functional modules. At first glance it seems like it would be easy to separate the modules by function as if they were “black boxes”. The initialization of global memory from file and terminal dialog is managed in the initialization module “a737int”. The equations of motion are mostly located in the module “eqmotn” as is the Dryden gust model. The Autopilot module “a737inp” comes next in the loop after the equations of motion module, and this is counterintuitive. All three autopilot systems, lateral, longitudinal, and thrust control are contained there. The control surface models modify the control parameter inputs of aileron, elevator, thrust, and rudder and drive the state space model contained in the differentiation and integration module “deriv\_ab2”. The automatic systems that control the stabilator, engine, and yaw damping of the rudder are part of the “deflect” module and not the autopilot module.

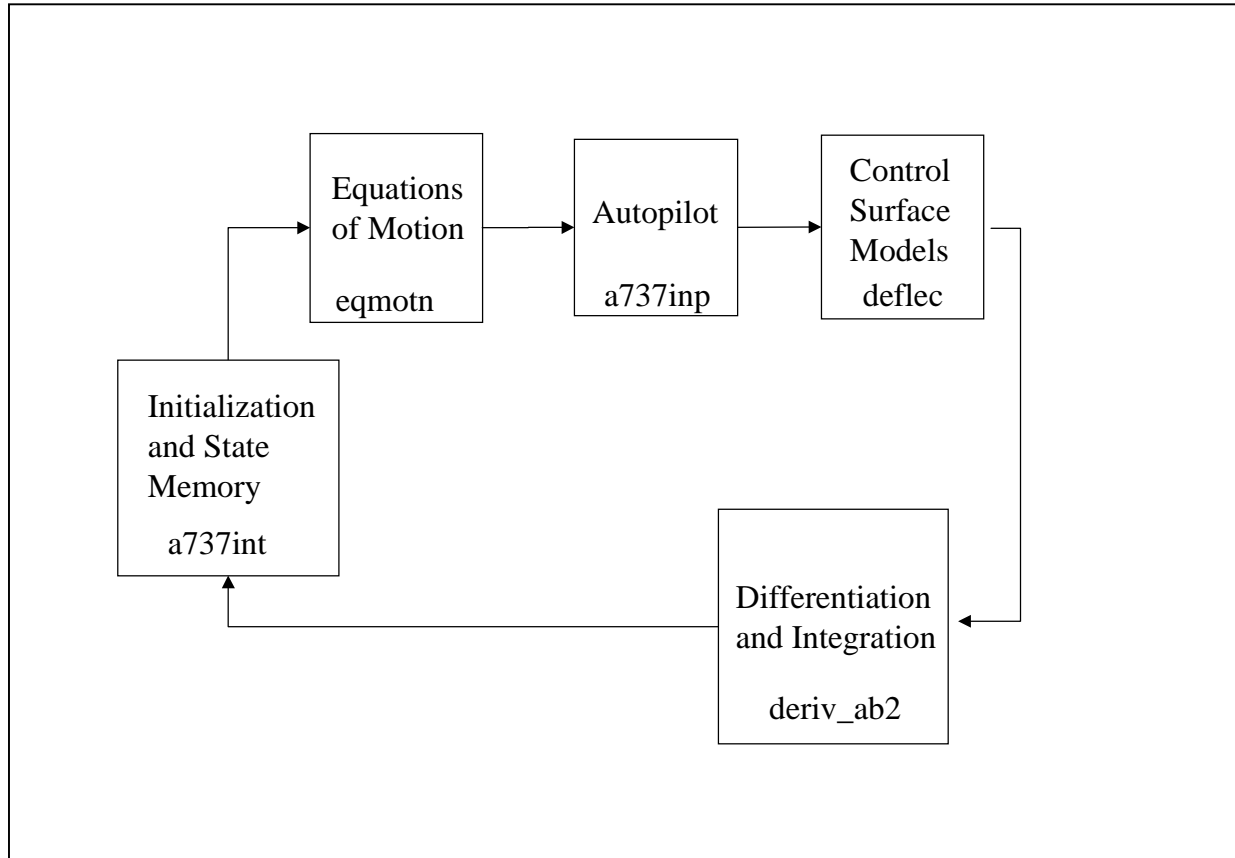


Figure 3. Data flow of the B-737 Autoland Simulation

The modules are coupled through subroutine calling sequences and global memory (FORTRAN common blocks). Data flow is hard to discern due to global memory linkage of subroutines and multiple names for the same global memory location through the use of the FORTRAN “equivalence” statement. The order of execution of the blocks is important because of the modification of the body reference frame velocities and rotations by the inertial reference frame accelerations in the “deriv\_ab2” module. First order lag filters and other modules use parameter values from the previous iteration to filter the current parameters.

The Simulink top level diagram is divided up into blocks similar to the FORTRAN model. There are more lines here because this is the actual source code diagram with all the variable linkages. Notice all the modules shown in the previous figure are present.

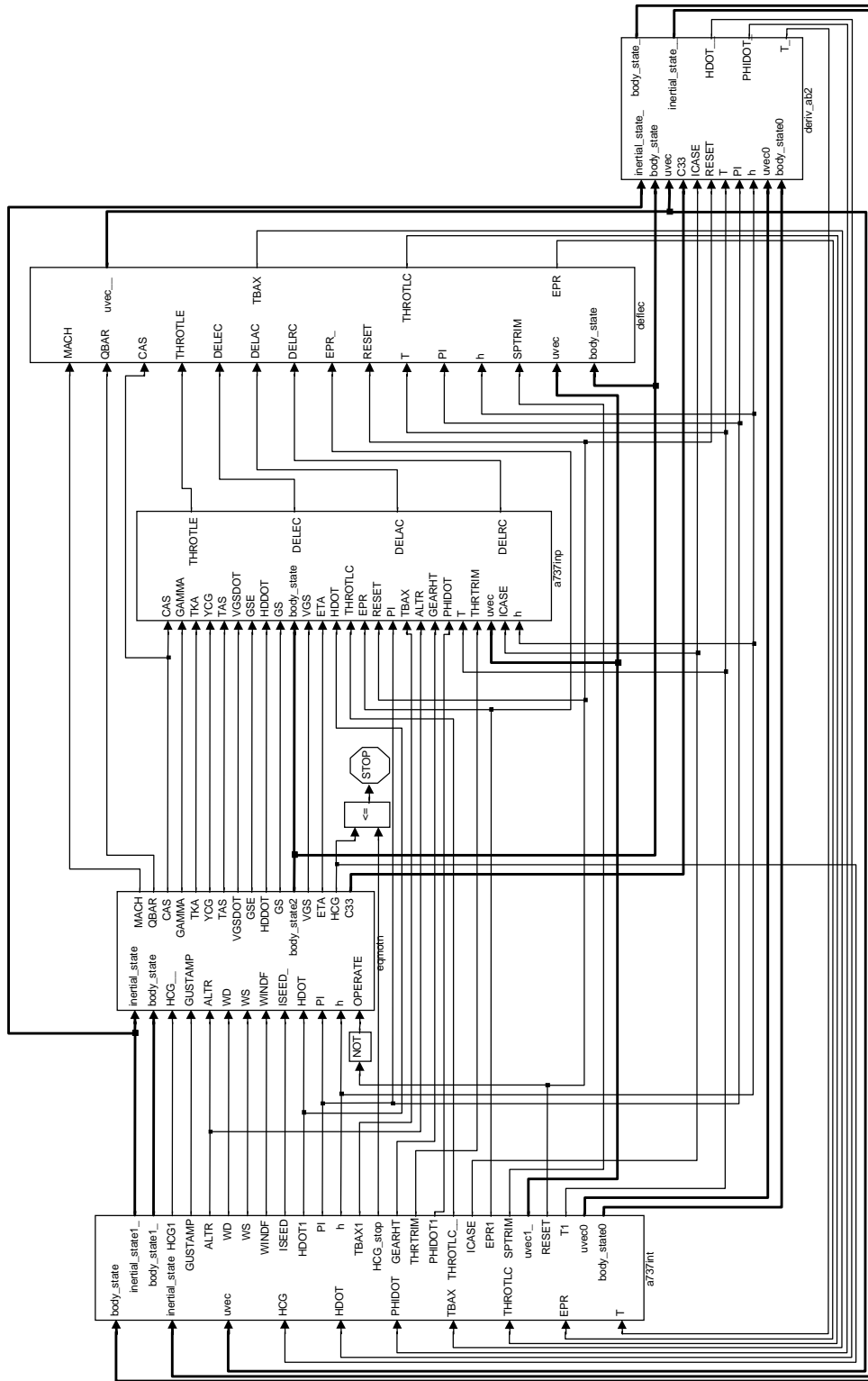


Figure 4. Major functional blocks appearing as a Simulink diagram.

## Newton's Laws of Motion and the Model

The FORTRAN and Simulink Autoland simulations both are based on the application of Newton's second law of motion  $F = ma$  for force, mass and acceleration. The three Cartesian components of motion come from this. The three rotational components of motion come from application of the rotational motion form of the second law,  $\tau = I\alpha$ . Torque replaces force, rotational inertia replaces mass, and rotational acceleration replaces Cartesian acceleration [Ref. 11].

Table 1. Motion Equations, Inertial Reference Frame

Rectilinear Motion		Rotation about a Fixed Axis	
Displacement	$x$	Angular Displacement	$\theta$
Velocity	$v = \frac{dx}{dt}$	Angular Velocity	$\omega = \frac{d\theta}{dt}$
Acceleration	$a = \frac{dv}{dt}$	Angular Acceleration	$\alpha = \frac{d\omega}{dt}$
Mass	$m$	Rotational Inertia	$I$
Linear Momentum	$p = mv$	Angular Momentum	$h = I\omega$
Force	$F = ma$	Torque	$\tau = I\alpha$
	$F = m \frac{dv}{dt}$		$\tau = I \frac{d\omega}{dt}$
	$F = \frac{dp}{dt}$		$\tau = \frac{dh}{dt}$

These six equations can be collected into two vector equations. The mass and acceleration terms can be expressed as derivatives of momentum and angular momentum.

## Force and Torque in the Inertial Reference Frame

Table 2. Motion Equations, Vector Notation

Rectilinear Motion		Rotation About A Fixed Axis	
Force Vector	$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$ $\mathbf{F} = \frac{d\mathbf{p}}{dt}$	Torque Vector	$\boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$ $\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$ $\boldsymbol{\tau} = \frac{d\mathbf{h}}{dt}$
Linear Momentum	$\mathbf{p} = m\mathbf{v}$	Angular Momentum	$\mathbf{h} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix}$ $\mathbf{h} = \mathbf{r} \times \mathbf{p}$
Position Vector	$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$		

## Body Reference Frame for Simplified Torque Calculation

The vector force and torque equations relate linear momentum to force and angular momentum to torque. The angular momentum is the cross product of the angular velocity and the rotational inertia. The angular velocity is referenced to rotation about the center of mass (gravity) of the aircraft relative to the orientation of axes fixed with respect to a beginning reference orientation. Translational velocity of this center of mass reference point is assumed to be not changing which makes it an inertial reference frame.

The position of a mass element in the body reference frame is given by:

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \quad (1)$$

The angular velocity of the body reference frame relative to the inertial reference frame is given by:

$$\boldsymbol{\omega} = P_b\mathbf{i} + Q_b\mathbf{j} + R_b\mathbf{k} \quad (2)$$

The velocity of a mass element  $dm$  in the inertial reference frame is:

$$\mathbf{v} = \mathbf{v}_{c.g.} + \boldsymbol{\omega} \times \mathbf{r} \quad (3)$$

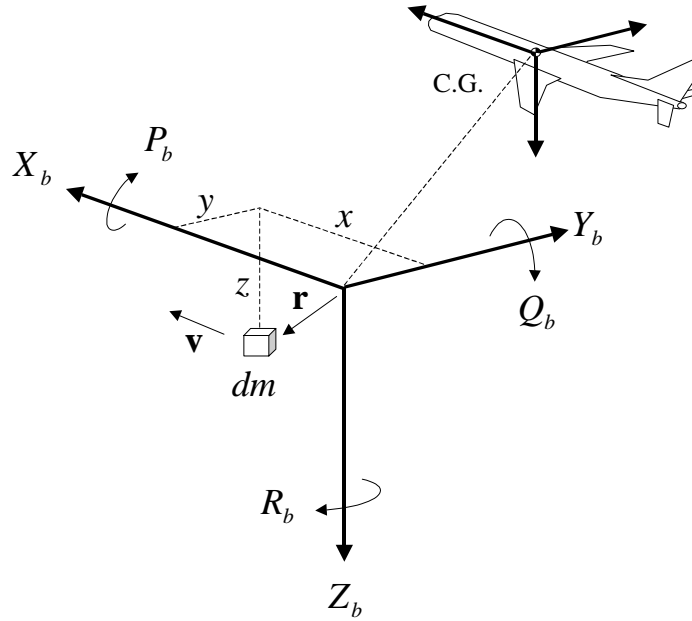


Figure 5. Moment of inertia coordinates.

The moment of inertia depends upon the spatial mass distribution of the aircraft and the orientation of the axes the moment is referenced to. It is a three by three matrix that changes as the inertial reference frame axes rotate relative to the principal axes of symmetry of the aircraft. If the reference frame axes are fixed to the aircraft, it ceases to be an inertial reference frame. Since this non-inertial reference frame rotates with the aircraft, the moment of inertia relative to this reference frame is simpler. The off-diagonal terms or products of inertia vanish. This is evident in the expressions for the angular momentum components relative to the earth-fixed inertial reference frame as opposed to the expressions for the angular momentum components relative to the body reference frame fixed in the rotating aircraft. The angular momentum components behave like the rotational inertia components under this transformation of axes.

The following development of equations of motion parallels an expanded version given in [Ref. 6, p 97-98]. Begin consideration of the angular momentum of the rigid body as being the sum of the angular momentum of the mass elements.

$$\mathbf{h} = \sum (\mathbf{r} \times \mathbf{v}) dm \quad (4)$$



Use equation (3)  $\mathbf{v} = \mathbf{v}_{c.g.} + \boldsymbol{\omega} \times \mathbf{r}$  to substitute for the velocity.

$$\mathbf{h} = \sum \mathbf{r} \times \mathbf{v}_{c.g.} dm + \sum \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm \quad (5)$$

$\mathbf{v}_{c.g.}$  is constant with respect to the summation and  $\sum \mathbf{r} dm = 0$ . This gives zero for the first term in equation (5) because  $(\sum \mathbf{r} dm) \times \mathbf{v}_{c.g.} = 0$ . The second term is expanded using the vector triple product expression.

$$\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) = \boldsymbol{\omega} r^2 - r(\boldsymbol{\omega} \cdot \mathbf{r}) \quad (6)$$

$$\mathbf{h} = \boldsymbol{\omega} \sum (x^2 + y^2 + z^2) dm - \sum \mathbf{r} (P_b x + Q_b y + R_b z) dm \quad (7)$$

Expanding the angular momentum vector in terms of its components and factoring to collect terms on  $P_b$ ,  $Q_b$ , and  $R_b$  results in:

$$\begin{aligned} h_x &= P_b \sum (y^2 + z^2) dm - Q_b \sum xy dm - R_b \sum xz dm \\ h_y &= -P_b \sum xy dm + Q_b \sum (x^2 + z^2) dm - R_b \sum yz dm \\ h_z &= -P_b \sum xz dm - Q_b \sum yz dm + R_b \sum (x^2 + y^2) dm \end{aligned} \quad (8)$$

Equation (8) is the angular momentum components with moment of inertia factors relative to a reference frame that is located at the airplane center of mass but does not rotate with the airplane, an inertial reference frame.

Looking at the angular momentum components relative to the body reference frame, which does rotate with the airplane, the cross terms become zero. This simplification comes at the expense of adding components of forces and torques that are an artifact of the reference frame being a rotating (accelerating, non-inertial) one.

$$\begin{aligned} h_x &= P_b \sum (y^2 + z^2) dm \\ h_y &= Q_b \sum (x^2 + z^2) dm \\ h_z &= R_b \sum (x^2 + y^2) dm \end{aligned} \quad (9)$$

Equation (9) is the angular momentum components with moment of inertia factors relative to the body reference frame aligned with the axes of symmetry of the aircraft. The Coriolis and centrifugal “fictitious forces” need to be accounted for by modification of the equations of motion. The next section describes the modifications necessary.

## Force and Torque in the Non Inertial Reference Frame of the Aircraft

The equations of motion fixed relative to the body reference frame of the aircraft rotate with an angular velocity  $\omega$  with respect to the inertial reference frame. The  $\omega$  term in the force equation is the part of the force resulting from the rotation of the frame of reference. It is a combination of centrifugal and Coriolis forces.

Force Equation 
$$\mathbf{F} = m \frac{d\mathbf{v}_{c.g.}}{dt} + m\omega \times \mathbf{v}_{c.g.} \quad (10)$$

Moment (Torque) Equation 
$$\boldsymbol{\tau} = \frac{d\mathbf{h}}{dt} + \omega \times \mathbf{h} \quad (11)$$

The force equation may be resolved into components along the non-inertial body reference frame. These components are given below.

$$\begin{aligned} F_x &= m(\dot{U}_b + Q_b W_b - R_b V_b) \\ F_y &= m(\dot{V}_b + R_b U_b - P_b W_b) \\ F_z &= m(\dot{W}_b + P_b V_b - Q_b U_b) \end{aligned} \quad (12)$$

The Simulink implementation of the force component equations is as shown in figure 6.

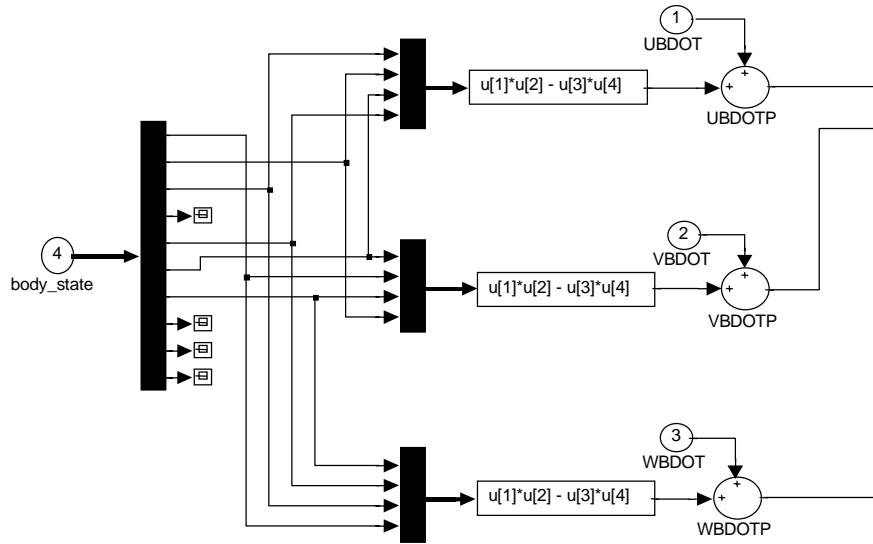


Figure 6. Force component calculation, Simulink implementation.

A similar transformation occurs with the torque components. The torque equation transformation is not found explicitly in the Simulink model but is taken care of implicitly in the transformation of the accelerations and velocities. Here is equation (11) repeated for convenience.

Moment (Torque) Equation 
$$\boldsymbol{\tau} = \frac{d\mathbf{h}}{dt} + \omega \times \mathbf{h} \quad (11)$$

These are the components of torque for the non-inertial reference frame. The  $L, M, N$  notation for the torque components is also presented.

$$\begin{aligned}\tau_x &= \dot{h}_x + Q_b h_z - R_b h_y, & \tau_x &= L \\ \tau_y &= \dot{h}_y + R_b h_x - P_b h_z, & \tau_y &= M \\ \tau_z &= \dot{h}_z + P_b h_y - Q_b h_x, & \tau_z &= N\end{aligned}\tag{12}$$

## Reference Frames

Several frames of reference are encountered when examining the FORTRAN and Simulink B-737 linear Autoland simulation. A body reference frame centered at the center of gravity (mass) of the airplane travels along with the vehicle and is subject to rotations and accelerations. It is an accelerated reference frame except when in straight-line motion. The stability reference frame X-axis points in the direction of motion of the airplane. To be consistent with the B-737 FORTRAN simulation variable nomenclature, stability and the body frame of reference will be considered to be the same. The inertial reference frame is attached to the center of the earth and is defined by latitude, longitude, and altitude of the center of gravity of the vehicle from center of the earth to locate its origin. It is oriented north and east. Euler angles are used to determine the orientation of the vehicle relative to the inertial reference frame. The runway reference frame is used to calculate the glide slope and localizer Cartesian position error during the automatic landing phase of the simulation.

## Body Reference Frame

The body reference frame is attached to the aircraft center of mass (gravity) and moves with the aircraft. The  $X_b$  axis points in the direction of motion, not necessarily along the aircraft's principal longitudinal dimension.

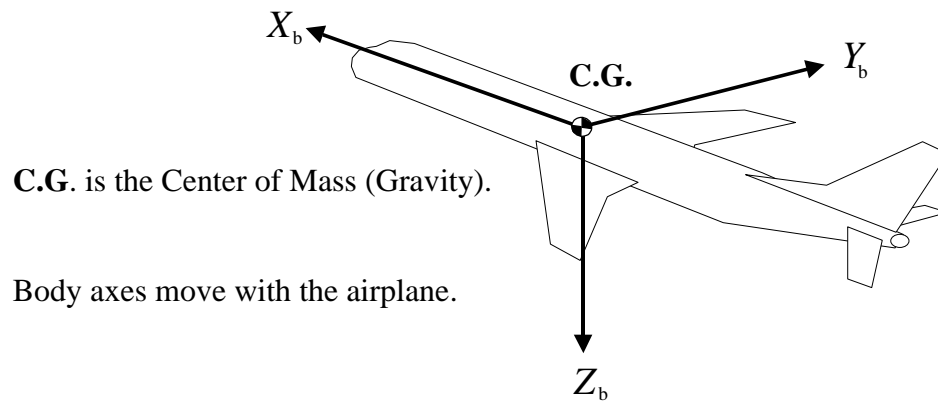


Figure 7. Body Cartesian axes.

The velocity components projected on the body reference frame are given by  $U_b, V_b, W_b$ . They are modified by pseudo-forces (centrifugal and Coriolis) that arise from the rotation of the coordinate reference frame since the body reference frame is not an inertial reference frame.

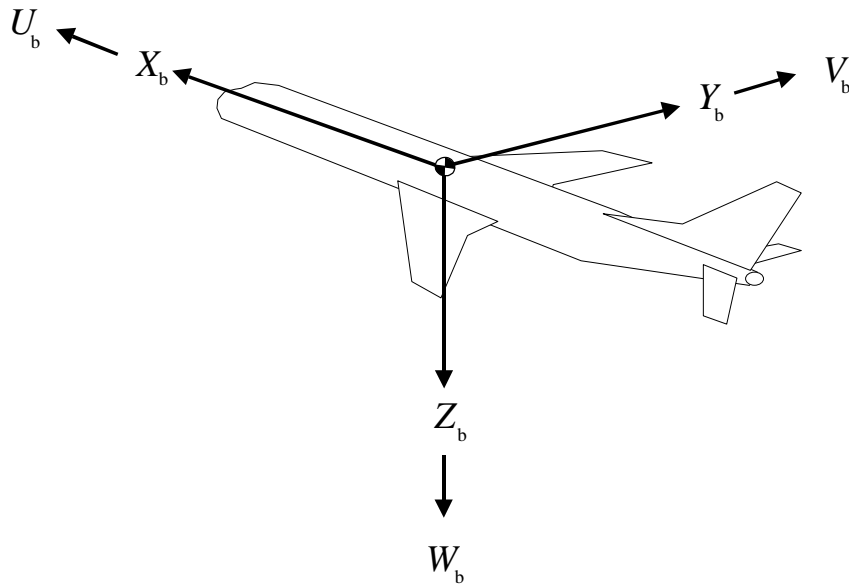


Figure 8. Body Cartesian velocity components  $U_b, V_b, W_b$ .

The rotation rates relative to the accelerated body reference frame are given by roll, pitch, yaw;  $P_b, Q_b, R_b$ . The Euler rotation angles are relative to the inertial reference frame.

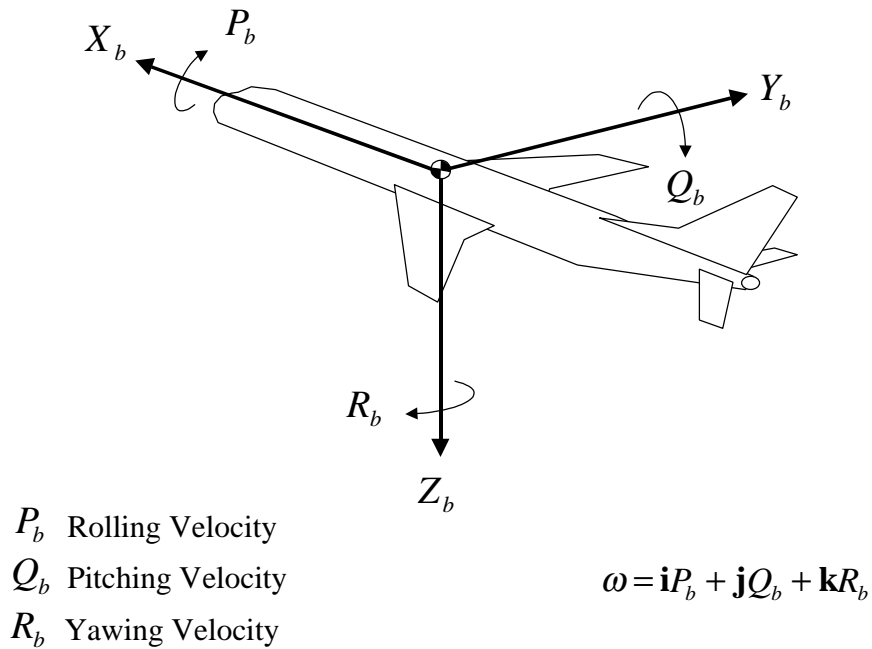


Figure 9. Body angular velocity components  $P_b, Q_b, R_b$ .

## Inertial Reference Frame

The origin of the inertial reference frame is given relative to the center of the Earth. The longitude angle in degrees west of Greenwich England defines one of the spherical coordinates. The latitude defines the angle above the equator. The sea level altitude of the aircraft center of mass (gravity) added to the mean radius of the earth locates the aircraft relative to the center of the earth.

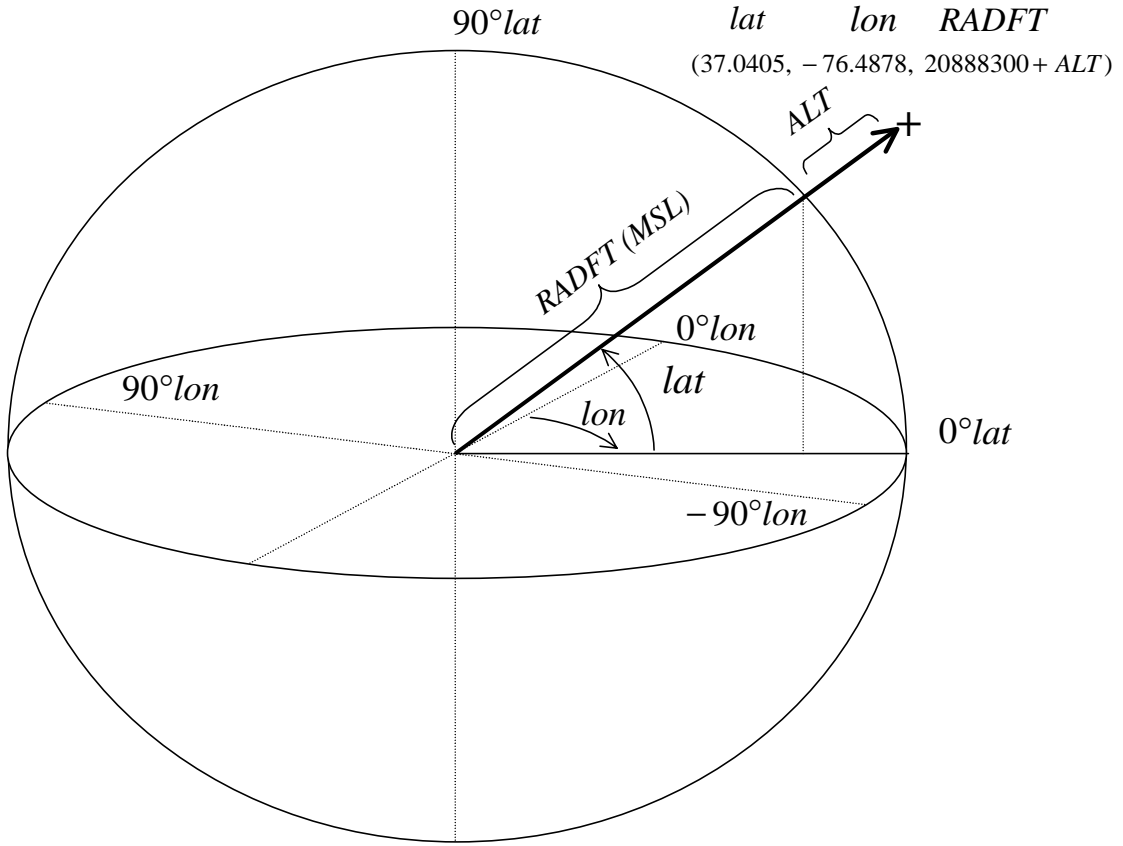


Figure 10. Inertial reference frame, radial view.

The inertial reference frame shown in this frame is the same as the reference frame shown in the next figure except this view shows the earth in cross section. This view of the globe is also rotated about the north-south axis. In this view  $X_{ref}$  points north,  $Y_{ref}$  points east, and  $Z_{ref}$  points down. Also, the starting condition for simulation flight case 1 has been shown with the  $X_0$  and  $Y_0$  axes rotated through the Euler yaw angle  $\psi_0$  clockwise from  $X_{ref}$  which faces north.

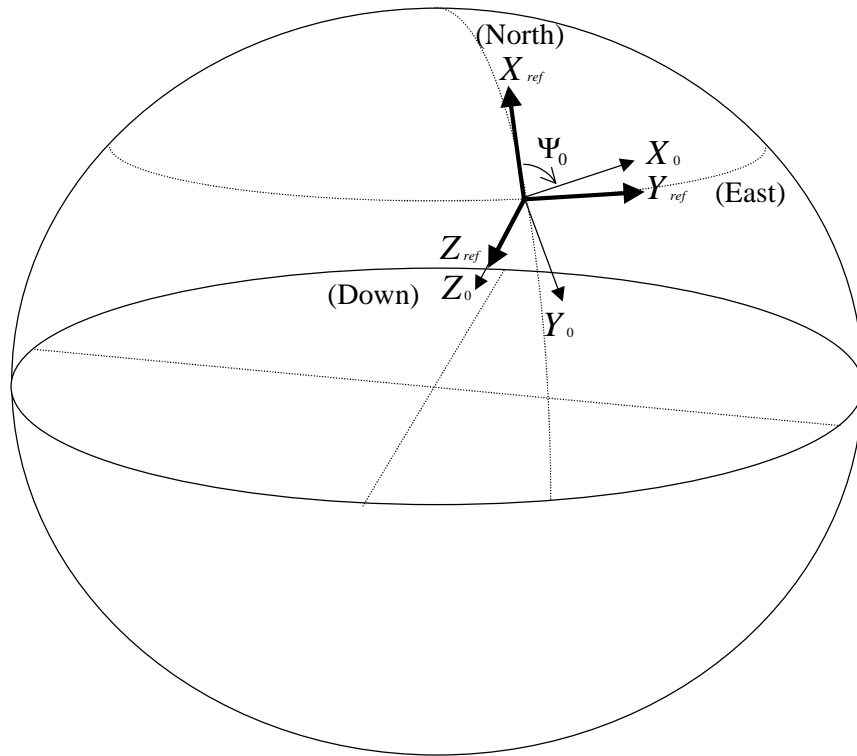


Figure 11. Inertial reference frame viewed from the earth's surface.

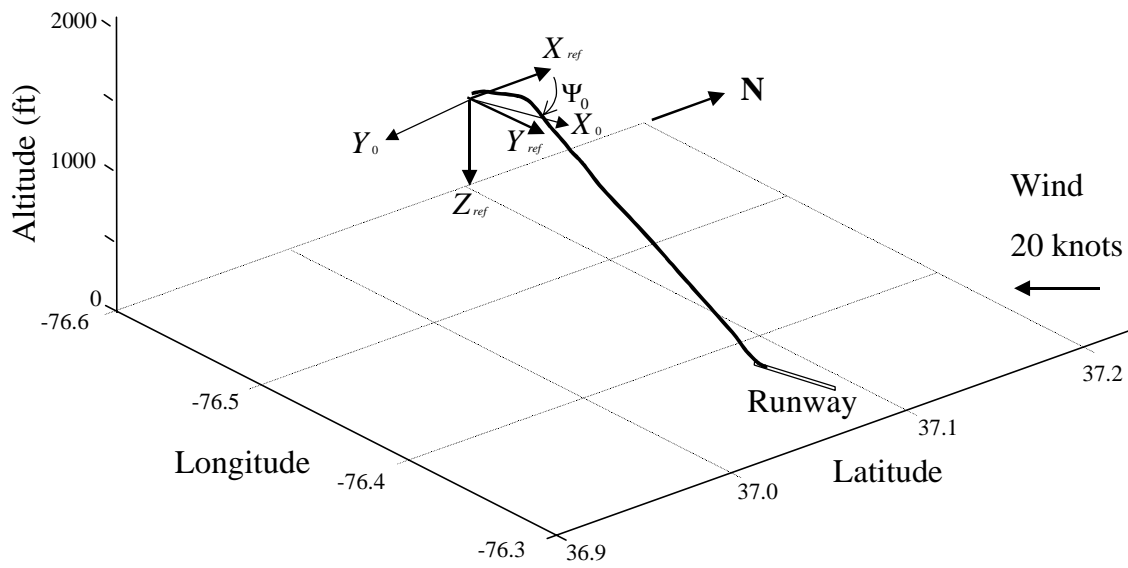


Figure 12. Inertial reference frame with flight case 1 and the runway shown from a point of view closer to the ground.

The time histories of  $Lat, Lon, Alt, \psi, \theta, \phi$  relative to the simulation origin reference axes  $X_{ref}, Y_{ref}, Z_{ref}$  defines the inertial position and orientation in figures 11 and 12. Figure 12 is a three dimensional Cartesian plot of the path of flight case 1 in latitude, longitude and altitude. Notice the orientation of north relative to figure 11.

### Orientation Using Euler Angles

#### Order of Rotation of Axes

<b>Yaw,</b>	<b>Pitch,</b>	<b>Roll</b>
$\Psi$	$\theta$	$\phi$
<b>Psi,</b>	<b>Theta,</b>	<b>Phi</b>

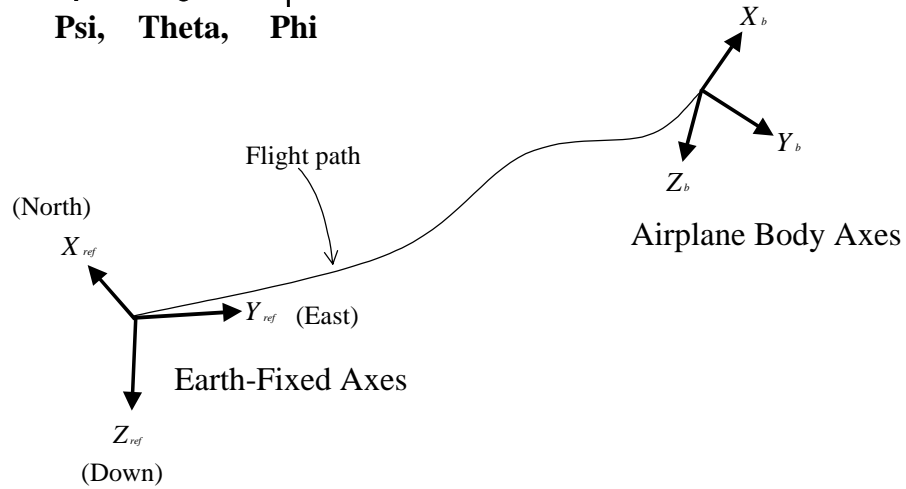
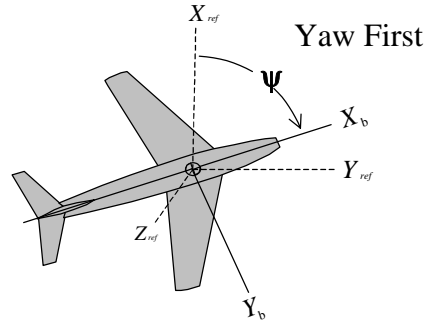


Figure 13. Airplane orientation using Euler angles.

Figure 13 depicts the orientation of the vehicle. Start with the reference axes pointing north and east. The reference axes are fixed relative to the earth and form an inertial reference frame. The rotation of the earth is neglected in the B-737 Linear Autoland simulation. All the orientation Euler angle time histories refer to the inertial frame defined at the beginning of the simulation. The aircraft orientation is specified by three successive rotations relative to the reference axes. First, yaw the aircraft according to the Euler angle  $\psi$ . Second, pitch the aircraft using Euler angle  $\theta$ . Third, roll the aircraft using Euler angle  $\phi$ . The order of the rotations follows this convention and is important. Another order of successive rotations will give an incorrect orientation. The aircraft illustration is from [Ref. 12].





## Order of Rotation of Axes

**Yaw, Pitch, Roll**  
 $\Psi$      $\theta$      $\phi$   
**Psi, Theta, Phi**

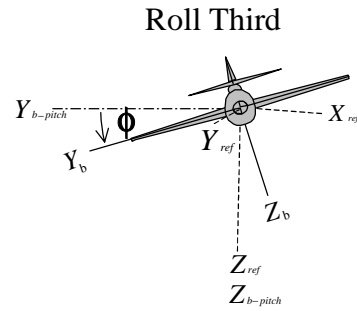
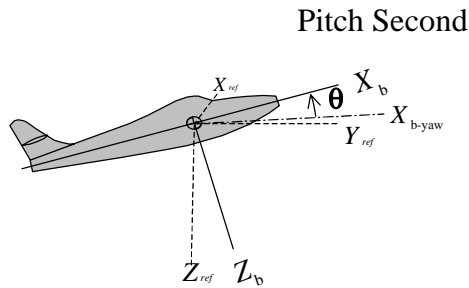


Figure 14. Sequence of rotations to define airplane orientation using Euler angles.

## Cartesian Velocity In Inertial Frame

The Cartesian velocity is transformed between the inertial reference frame fixed to earth at the starting point and the body reference frame that travels with the airplane center of mass (gravity). The inertial reference frame velocity components are given in the following table.

Table 3. Notation for the Inertial Reference Frame Cartesian Velocity Components

Inertial Reference Frame Axis	Inertial Velocity	Name in Simulink Model
$X_{ref}$	$V_N$	$VN$
$Y_{ref}$	$V_E$	$VE$
$Z_{ref}$	$V_D$	$VD$

The transformation used to convert the velocities in the inertial reference frame to the body reference frame is the formula reproduced here for converting a coordinate frame oriented using Euler angles [Appendix A, page 67, Gainer]. The notation shown is as given in the reference. For our purposes, replace  $X', Y', Z'$  with  $U_b, V_b, W_b$  and replace  $X, Y, Z$  with  $V_N, V_E, V_D$ .

The transformation from the initial system to the primed system aligned at Euler angles with the initial system is

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \Gamma \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (13)$$

$$\Gamma = \begin{bmatrix} \cos\theta \cos\psi & \cos\theta \sin\psi & -\sin\theta \\ \sin\phi \sin\theta \cos\psi - \sin\psi \cos\phi & \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi & \sin\phi \cos\theta \\ \cos\psi \cos\phi \sin\theta + \sin\psi \cos\phi & \sin\psi \cos\phi \sin\theta - \cos\psi \sin\phi & \cos\phi \cos\theta \end{bmatrix}. \quad (14)$$

This is the Euler angle forward coordinate transformation.

The inverse transformation from the primed system to the initial system aligned at Euler angles with the initial system is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (15)$$

$$\Gamma^{-1} = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \sin\psi \cos\phi & \cos\psi \cos\phi \sin\theta + \sin\psi \cos\phi \\ \cos\theta \sin\psi & \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi & \sin\psi \cos\phi \sin\theta - \cos\psi \sin\phi \\ -\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix}. \quad (16)$$

This is the Euler angle inverse coordinate transformation.

Notice that the inverse transformation matrix  $\Gamma^{-1}$  is the transpose of the forward transformation matrix. Considering the Euler angles as constants within a simulation time step, the elements of the same matrix can be taken in different order to make the inverse transformation [Ref. 6, p 35, equation (2.18)].

The following fragment of the equations of motion block diagram eqmotn1 shows the implementation of the  $\Gamma$  Euler angle forward coordinate transformation as implemented in Simulink. Notice the  $\phi, \psi, \theta, V_N, V_E, V_D$  Euler angle inputs, inertial velocity inputs, and the  $U_b, V_b, W_b$  body velocity outputs. The sine and cosine expressions are contained in the “b\_to\_inertial” block. The output elements of the transformation matrix from the “b\_to\_inertial” block are sent to other modules where a coordinate transformation is required. Simulink vector C33 is used to transfer the elements.

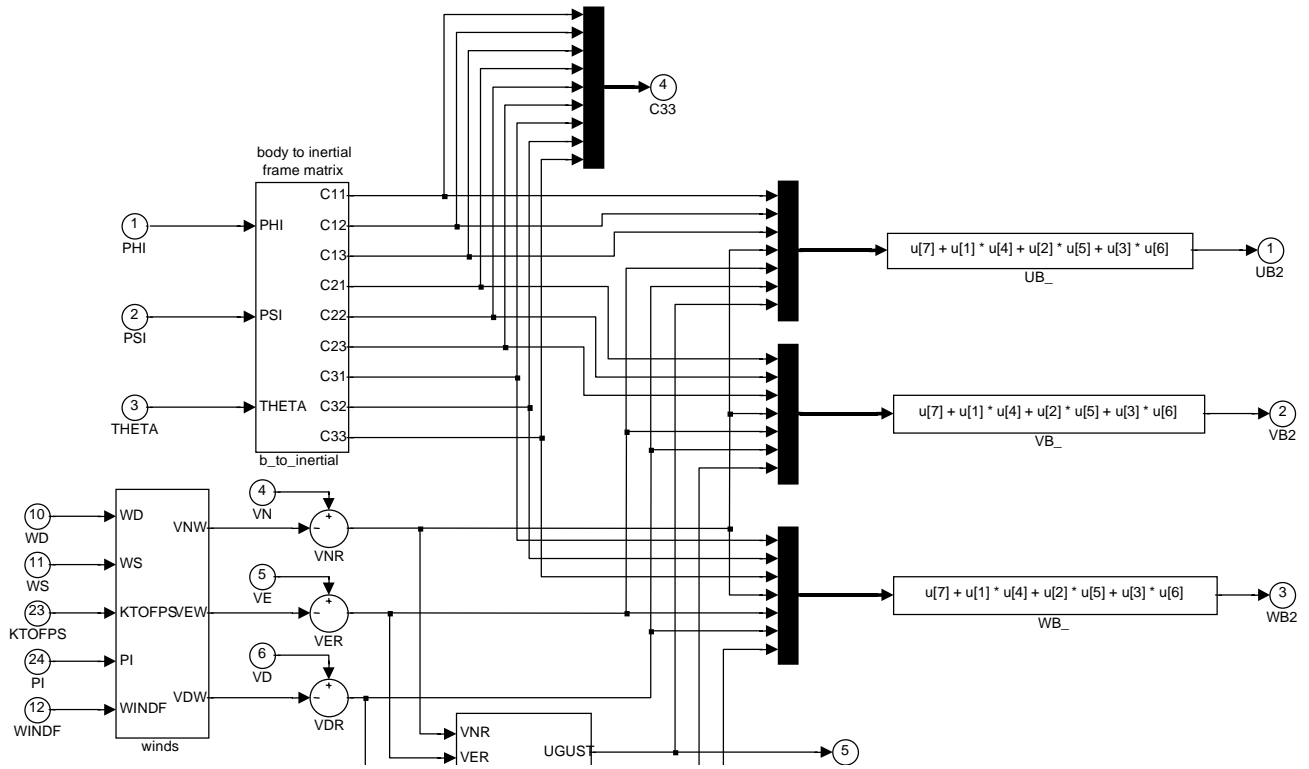


Figure 15. Euler angle transformation, inertial to body frame, Simulink implementation.

For the  $\Gamma^{-1}$  inverse Euler angle coordinate transformation, the Simulink vector,  $C33$ , is picked apart in such a way that the transpose of the transformation matrix results. It is used here to transform velocity derivatives (accelerations) from the body frame to the inertial reference frame. The following is a fragment from the “inertial accel” block inside the block “deriv”. The body frame velocity derivatives  $UBDOT, VBDOT, WBDOT$  are transformed to become inertial frame velocity derivatives  $VNDOT, VEDOT, VDDOT$ .

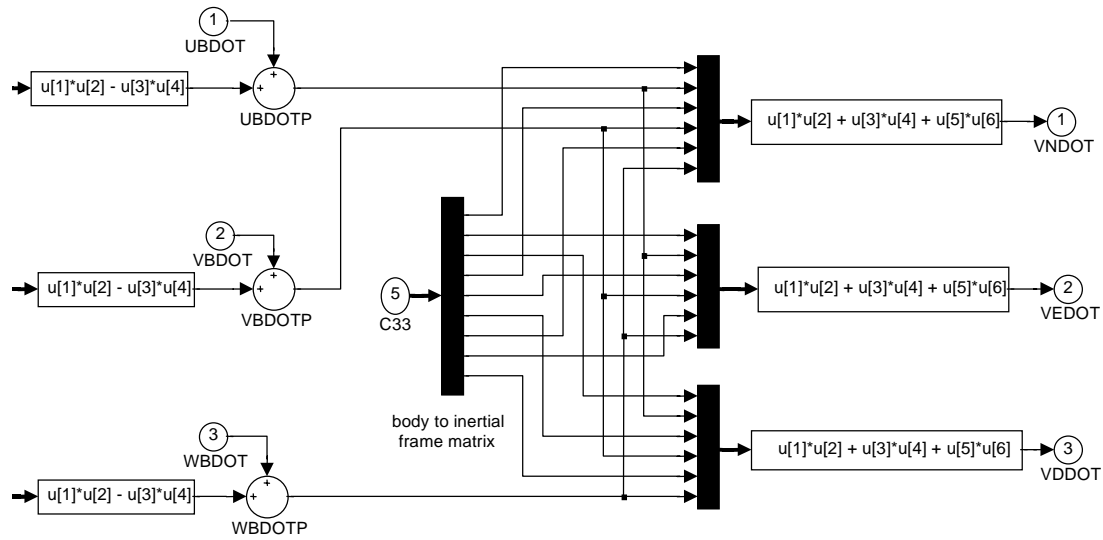


Figure 16. Euler angle inverse transformation, body to inertial frame, Simulink implementation.

### Angular Velocity In Inertial Frame

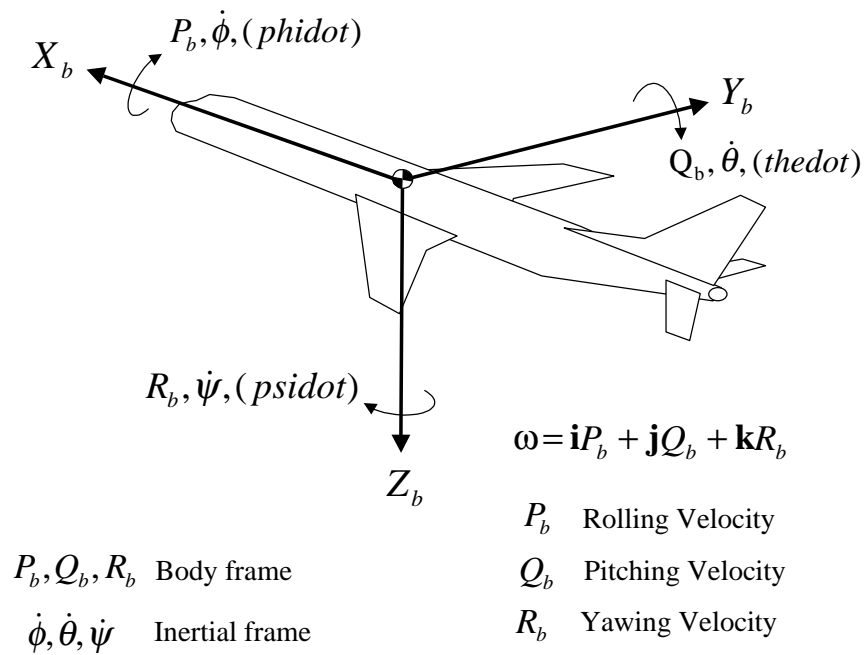


Figure 17. Angular velocity components.

Angular velocity in the inertial frame of reference is related to the angular velocity in the body frame according to the following transformation equations from [Ref. 6, p 102, equation (4.5,3)].

$$\begin{aligned} P_b &= \dot{\phi} - \dot{\psi} \sin \theta \\ Q_b &= \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \\ R_b &= \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi \end{aligned} \quad (17)$$

For the special case where the Euler axes are lined up with the body axes, the rotational velocities agree across the frames. The figure shows this special case. It is not that way in the model generally. The above equations are not seen in the model because the transformation of the Cartesian velocities takes care of this.

### Autopilot Runway Frame

The coordinate frame centered on the runway threshold is used during the landing phase. The  $X_{loc}$  values are negative until the threshold is crossed. This reference frame is used in the calculation of glide slope error and localizer error.

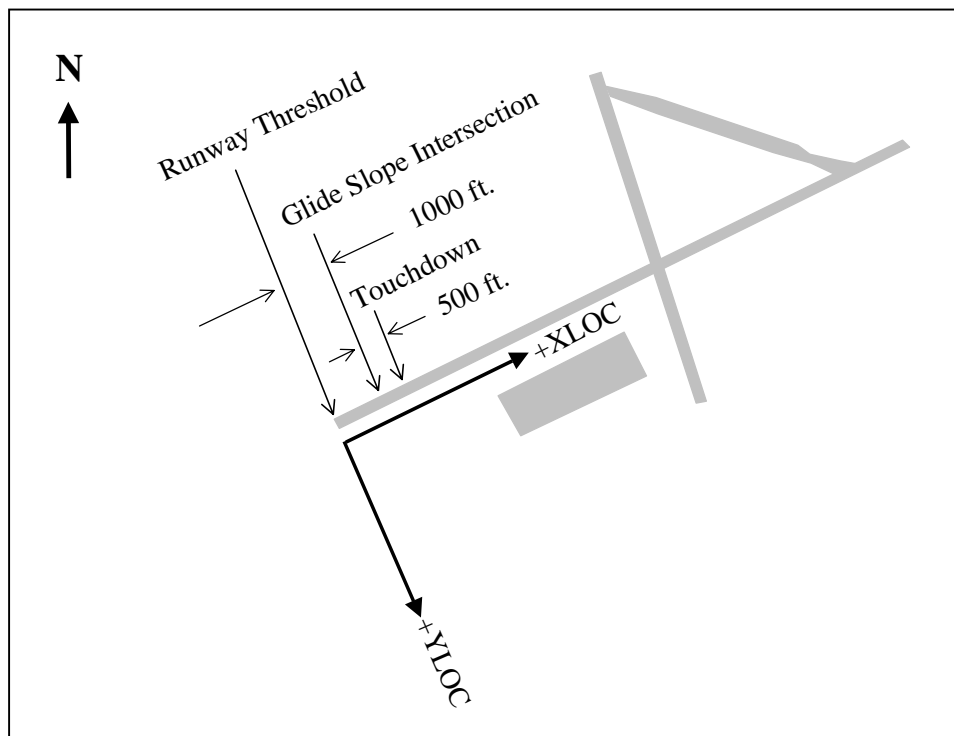


Figure 18. Runway reference frame.

## State Space Model

The following outline of state variable analysis can be found in more detail in [Ref. 10, chapter 7]. The output of a simple continuous system may be described by a transfer function acting on the input. The transfer function represents an  $n$ th order differential equation that relates output to input. There are  $n$  initial conditions on the dependent variable necessary to solve this equation. This set of initial conditions, represented as the components of vector  $\mathbf{x}$  defines a state of the system and is necessary to find the future response. At a starting time  $t_0$ , instead of an  $n$ th order differential equation,  $n$  simultaneous first order differential equations can approximate the system response. These act on  $\mathbf{x}$  to produce a vector of derivatives of the components of  $\mathbf{x}$ , represented as  $\dot{\mathbf{x}}$ . The coefficients of the first order equations can be collected in a matrix  $\mathbf{A}$ . This is called the dynamics matrix of the system.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \quad (18)$$

A system can also have the output modified by one or more independent inputs or forcing functions. The independent inputs form the components of a control-input vector  $\mathbf{u}$ . The coefficients of the inputs collected in a matrix  $\mathbf{B}$  can be added to the previous equation. Matrix  $\mathbf{B}$  is called the control matrix of the system.

$$\begin{array}{ccc} \text{State Vector} & \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} & \text{Control Inputs} & \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} \\ & & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} & (19) \end{array}$$

$$\mathbf{x} = \int \dot{\mathbf{x}} dt \quad (20)$$

The integral of the equation for  $\dot{\mathbf{x}}$  gives the state of the system at any time  $t$ . Figure 19 from [Ref. 10, p 251] is a simplified representation of the state approach for a state vector with two components.

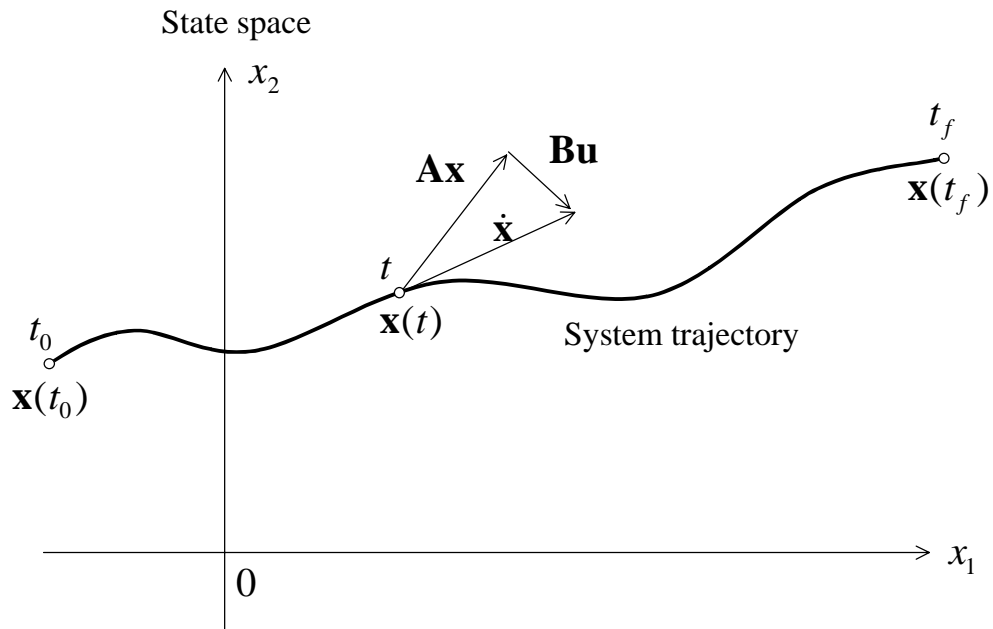


Figure 19. A state space trajectory with two independent variables.

Figure 20 shows the relationship in sequence between the  $\mathbf{A}$  and  $\mathbf{B}$  coefficient matrices from [Ref. 10, p 252]. The vector of control inputs,  $\mathbf{u}$ , the vector of state derivatives,  $\dot{\mathbf{x}}$ , and the  $n$  simultaneous equations represented by the  $n$  parallel integrators produce a state vector  $\mathbf{x}$ .

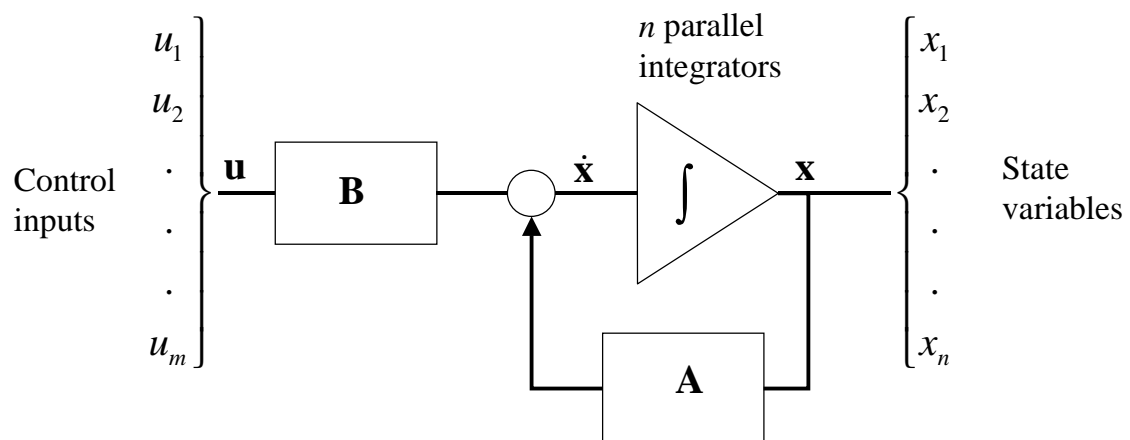


Figure 20. Data flow for a state space model.

## State Vector Description

Figure 21 defines the components of the body state vector. The roll, pitch, and yaw Euler angles are considered body state parameters even though they are measured relative to the inertial reference frame fixed at the model starting point.

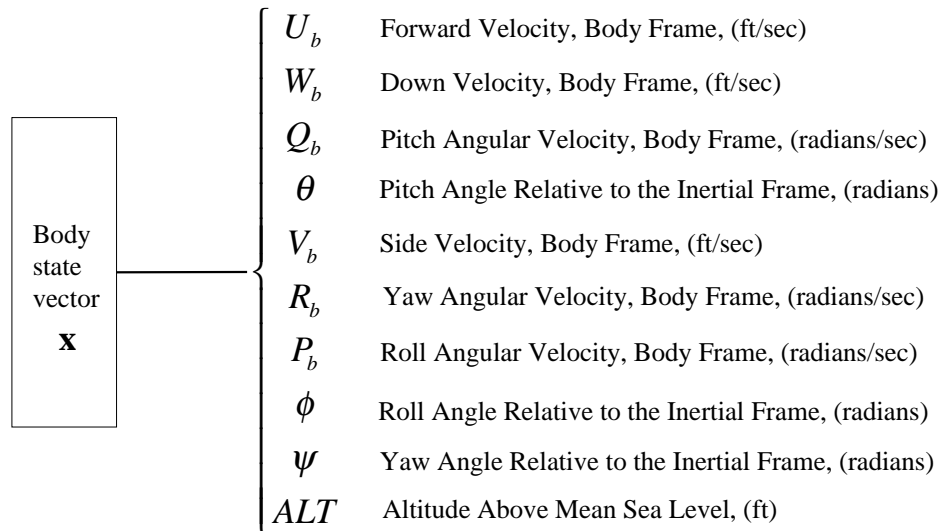


Figure 21. Body state vector components.

## ***FORTTRAN Common for State Vector***

In the original Linear B-737 Autoland Simulation for Airlabs FORTRAN code, a key data structure is the global shared memory. The shaded portions show the body state vector and the derivative of the body state vector. Notice the “equivalence” statement that gives two names to the state vector variables. This is the heart of the original FORTRAN model. The differentiation and integration modules “deriv” and “ab2” use the “RSTATE” representation to iterate across all components of the body state vector in order to differentiate and integrate the simulation to get the next state value for the next time step.



```

COMMON / CAB2 /
*
*      T      , H      , UB      , WB      ,
*      QB      , THETA  , VB      , RB      ,
*      PB      , PHI    , PSI    , ALT    ,
*      LAT     , LON    , VN     , VE     ,
*      VD      , UBDOT  , WBDOT  , QBDOT  ,
*      THEDOT  , VBDOT  , RBDOT  , PBDOT  ,
*      PHIDOT  , PSIDOT , HDOT   , DLAT   ,
*      DLON    , VNDOT  , VEDOT  , VDDOT
C
      DIMENSION RSTATE(10), RSTATED(10)
C
      REAL * 8 LAT, LON, VN, VE, VD, DLAT, DLON, VNDOT, VEDOT, VDDOT
C
      EQUIVALENCE (RSTATE(1), UB)
      EQUIVALENCE (RSTATED(1), UBDOT)

```

Figure 22. FORTRAN common block for state vector  $\mathbf{x}$  and its derivative  $\dot{\mathbf{x}}$ .

### Control Input Vector Description

This gives the definition of the components of the control input vector to the state space model. The aileron deflection and spoiler deflections come from the roll control system. The elevator deflection comes from the pitch control system. Both are actuator models. The rudder deflection comes from the yaw damper system. The stabilator position comes from the automatic stabilator trim system. The thrust comes from the engine model. The actuator models will be addressed in a subsequent section.

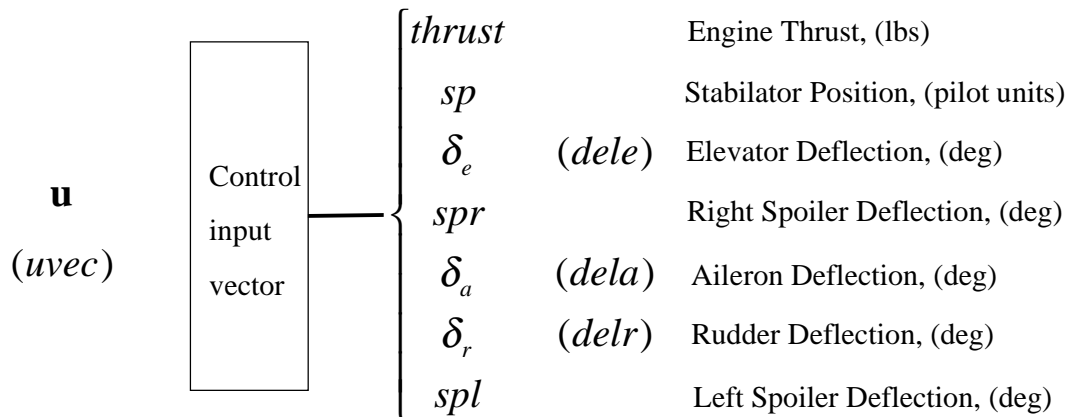


Figure 23. Control input vector components.

## ***FORTTRAN Common for Control Input Vector***

Like the state vector, the shaded portions show the control input vector in the Airlab FORTTRAN model. Notice the “equivalence” statement that gives two names to the control vector variables. The differentiation module “deriv” uses the “UVEC” representation to iterate across all components of the control input vector in order to differentiate the state vector to get the next state derivative value for the next time step. The primary control inputs to the state space model listed here are not those that come out of the autopilot modules. Commanded throttle THROTLC, commanded elevator DELEC, commanded aileron DELAC, and commanded rudder DELRC come from the autopilot modules and are modified by the actuator models or other trim control systems like the stabilator trim or the yaw damper systems. There is a hierarchy of subsystems that surrounds the state space model.

```

COMMON / CDEFLEC /
*
*
*
C
    DIMENSION UVEC(10)
C
    EQUIVALENCE (UVEC(1), THRUST)
    THRUST , SP , DELE , SPR ,
    DELA , DELR , SPL , UVEXTRA(3) ,
    GEAR , FLAPS , E63 , E36 ,
    SBHP , EPR , THROTLC

```

Figure 24. FORTTRAN common block for control vector  $\mathbf{u}$ .

## **Inertial State Vector**

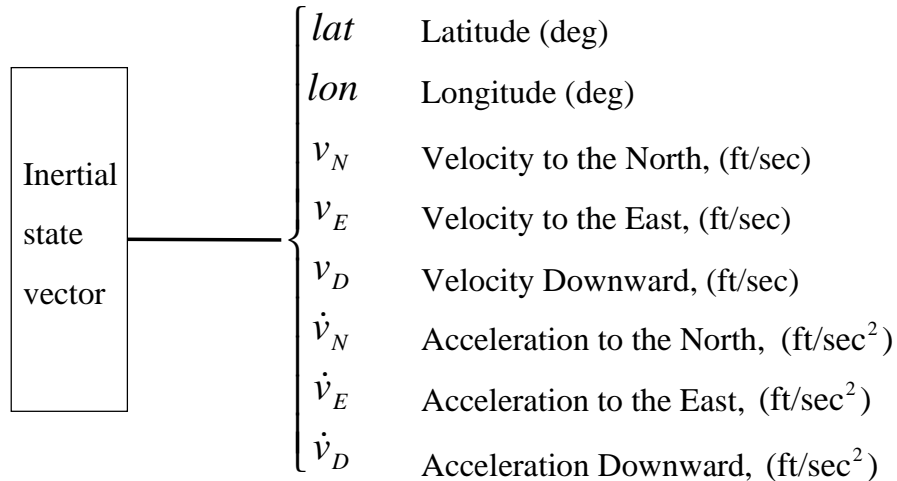


Figure 25. Inertial state vector components.

The inertial reference frame parameters are collected in a vector called the “inertial state vector”. This corresponds to the body state vector, except through an Euler angle transformation.

The calculation of the inertial accelerations from the body velocity derivative is shown in the state space derivative Simulink diagram. The “case1\_abmat” block is the output of the state space model for flight case 1. There are two state space model blocks because different coefficients were hard-coded for the **A** and **B** dynamics and control matrices for the two flight cases. The latitude and longitude calculation is shown in the lower left corner. These equations are based on the spherical earth model as opposed to the WGS84 ellipsoidal earth model currently in use by the aviation industry. The north component of velocity is divided by the sum of the earth radius and the aircraft altitude to get latitude. The east component of velocity is treated similarly but also scaled by latitude to account for the variation in the arc distance between degrees of longitude with distance from the equator.

$$lat = \frac{v_N}{R_{earth} + alt} \quad (21)$$

$$lon = \frac{v_E}{(R_{earth} + alt) \cos(lat)} \quad (22)$$

### ***Simulink Implementation of the State Space Model***

The state space model is implemented in the Simulink block “deriv\_ab2”. The **A** and **B** matrices are multiplied by the control input vector and the body state vector. The coefficients are multiplied column by column because of restrictions on passing arrays in the version of Real-Time Workshop used. The state space model output is transformed using the Euler angle inverse transformation  $\mathbf{\Gamma}^{-1}$  into the inertial reference frame to calculate the north, east, and down velocities. The north and east velocities are used to calculate the derivatives of latitude and longitude.

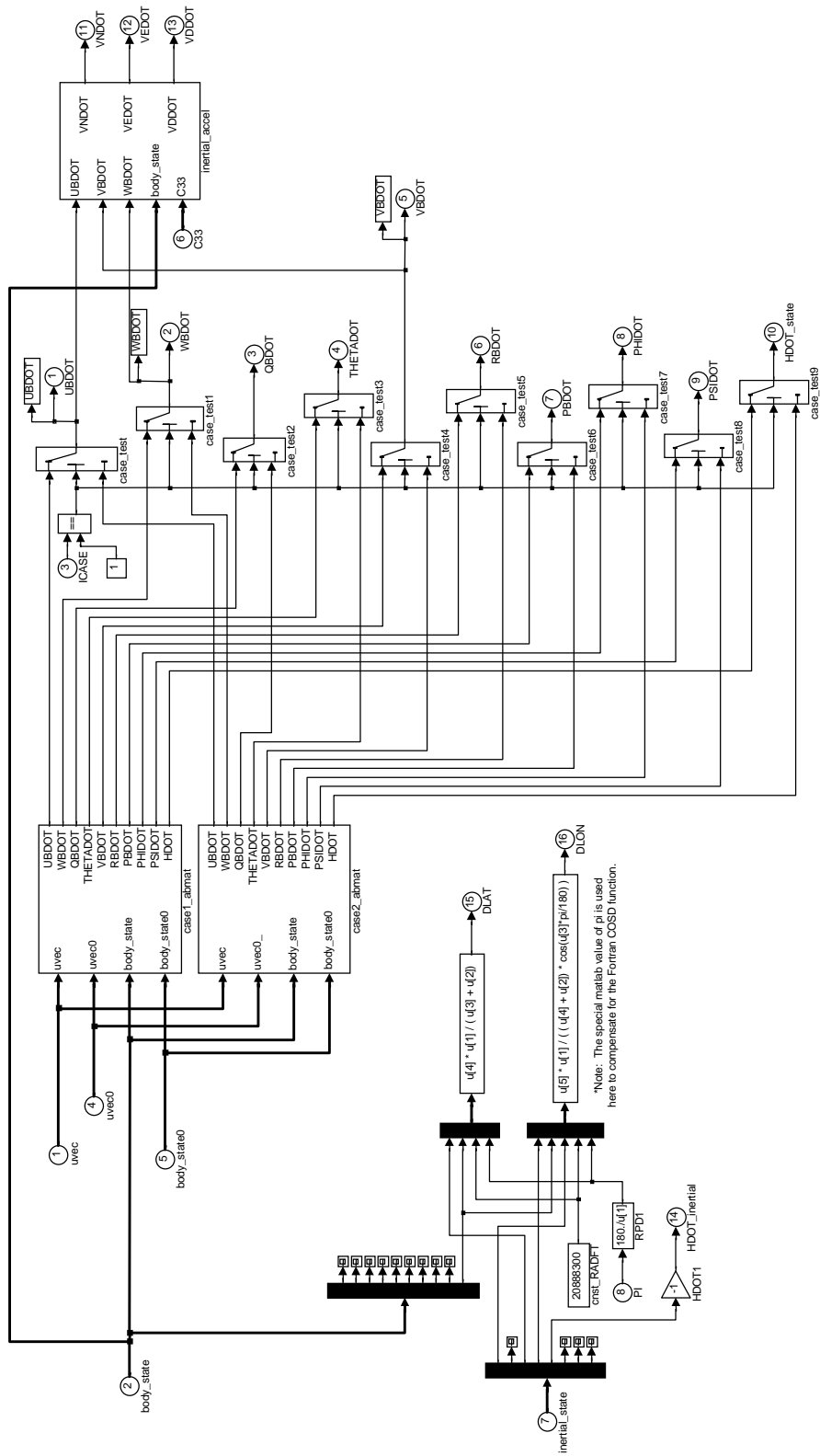


Figure 26. State space derivative Simulink diagram.

The latitude, longitude, north, east, and down velocities are part of the inertial reference frame description of the airplane state. The inertial velocities are not part of the state matrix simultaneous first order differential equations as defined in the B-737 Autoland model. The Newtonian laws of motion that relate change in momentum to force on a body assume an inertial reference frame (one that is not accelerated or rotating). Calculations are made with respect to an inertial reference frame and the non-inertial body reference frame. It is more convenient to calculate moments of inertia from non-inertial body reference frame as shown in the section on simplified torque calculation.

The body reference frame is not an inertial reference frame and exhibits fictitious forces due to its acceleration. The fictitious forces that are an artifact of rotation are the centrifugal and Coriolis forces. The laws of motion need to be modified to account for these. A side calculation uses the output of the body state vector to calculate the body acceleration components from the body velocity derivative. These are converted to acceleration components in the inertial reference frame. These inertial acceleration components are integrated to get inertial velocities for the next time step. The velocities are integrated to get inertial position coordinates, latitude and longitude, for the next time step. Finally, the inertial velocities are transformed into body velocities using the Euler angle coordinate transformation  $\mathbf{T}$ . In this way, the centrifugal and Coriolis forces are accounted for. This is what makes the order of calculation important.

## Equations of Motion

The equations of motion module contains four component modules. Module “eqmotn1” contains the transformation of velocity from the inertial reference frame to the body reference frame using the Euler angle transformation matrix. The effect of winds and gusts is applied in this block as well. The Dryden gust model contained here will be discussed in a subsequent section.

The module “eqmotn2” calculates the angle of attack  $\alpha$ , sideslip angle  $\beta$ , various airspeeds, atmosphere state parameters, and Mach number. The angle of attack and sideslip angle are a simple trigonometric calculation based on body velocity components. The atmosphere state parameters come from a polynomial function of altitude [Ref. 21]. Density, temperature, pressure and speed of sound are obtained in this way. Total body velocity, true air speed, and calibrated airspeed are calculated from the body velocity components and Mach number corrections. The following contains definitions of the different air speeds as shown in [Ref. 2, p 4-6].

Table 4. Air Speed Definitions

Airspeed	Meaning of Name	Definition
IAS	Indicated Airspeed	Measured using a pitot tube and static port to get the difference between total pressure and static pressure. The pilot's instrument is calibrated to read this.
CAS	Calibrated Airspeed	The measured pitot and static port readings are corrected to account for the pitot tube being not aligned with the airstream at a non-zero angle of attack. Other instrumentation and installation errors are accounted for.
TAS	True Airspeed	Actual speed at which the aircraft moves through the air. CAS is corrected for non standard temperature and pressure.
MACH	Mach Number	Ratio of true airspeed to the speed of sound.

### Localizer and Glide Slope Error

The module “eqmotn3” calculates the position of the aircraft relative to the runway reference frame to support the instrument landing system (ILS) for the Autoland. The latitude, longitude, altitude, earth radius, and runway position coordinates are used to calculate the position of the aircraft in the runway reference frame. The aircraft center of mass (gravity) position components in the runway frame are  $(X_{cg}, Y_{cg}, H_{cg})$ . Simple trigonometry is used to calculate the glide slope error  $GSE$  and the track error  $ETA$  for the ILS.

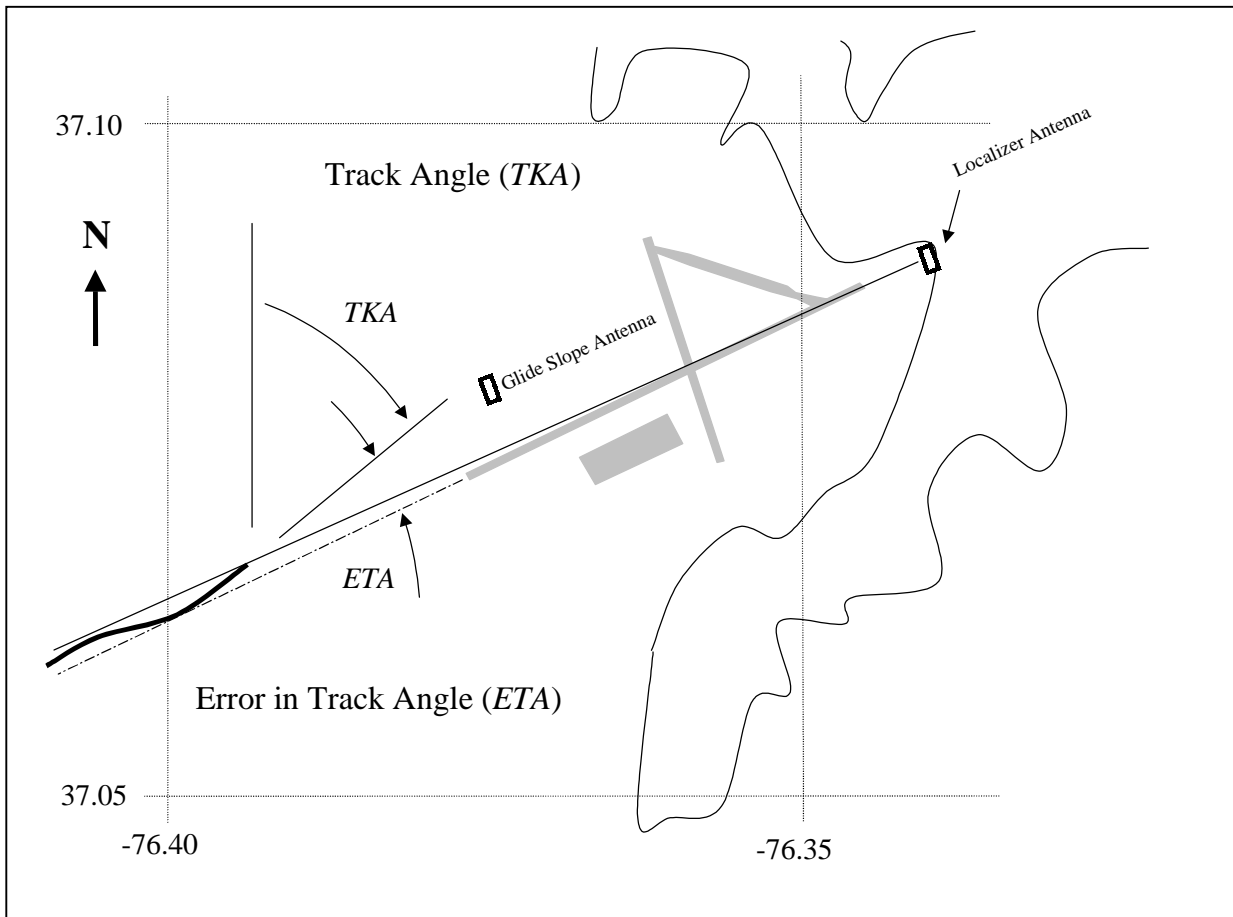


Figure 27. Localizer antenna and error in track angle geometry.

Module “eqmotn4” calculates the ground speed, track angle, and flight path angle. Ground speed is the projection of the true airspeed (*TAS*) on the ground. Winds cause ground speed to differ from true airspeed. The flight path angle  $\gamma$  is the angle of descent. It should be nearly the same as the glide slope during descent. The difference between flight path angle and glide slope is exaggerated to distinguish between the two.

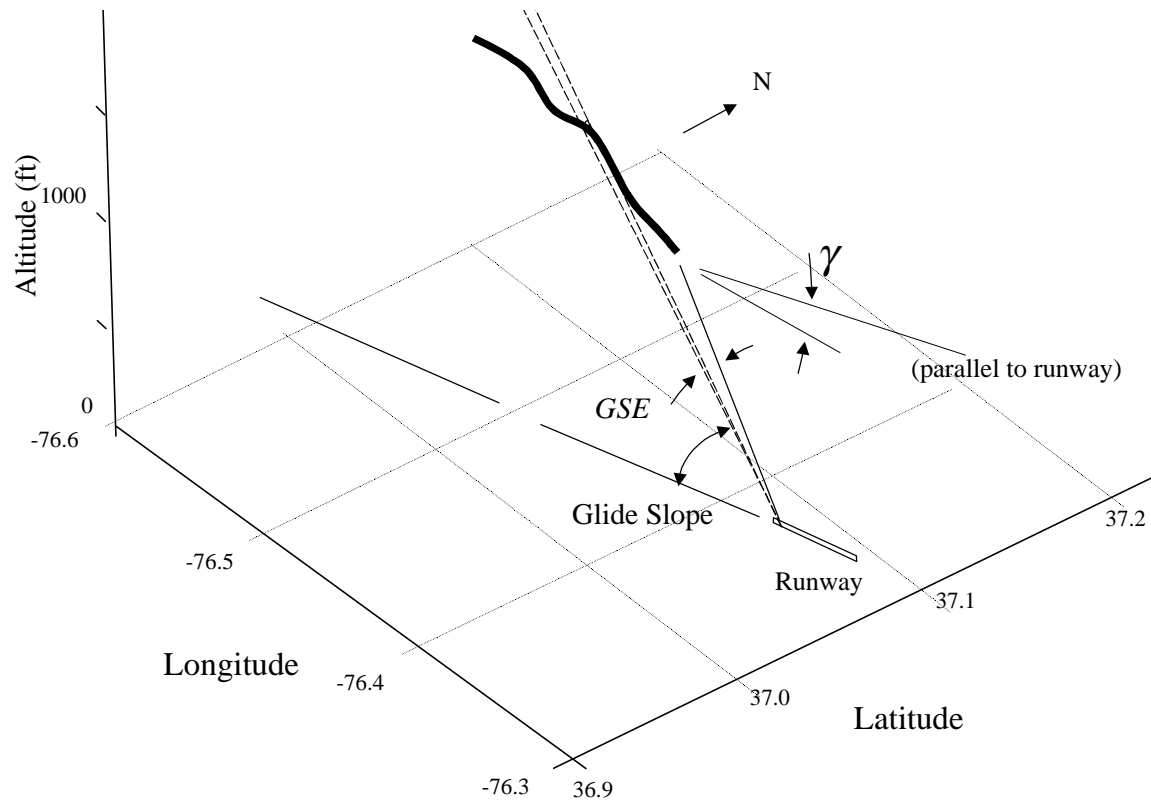


Figure 28. Glide slope error (*GSE*) and flight path angle  $\gamma$  .



## Dryden Gust Model

The Dryden gust model is based on empirically measured power spectra of wind velocity in turbulent air. Two plots of the power spectrum measured by researchers [Ref. 16] are shown.

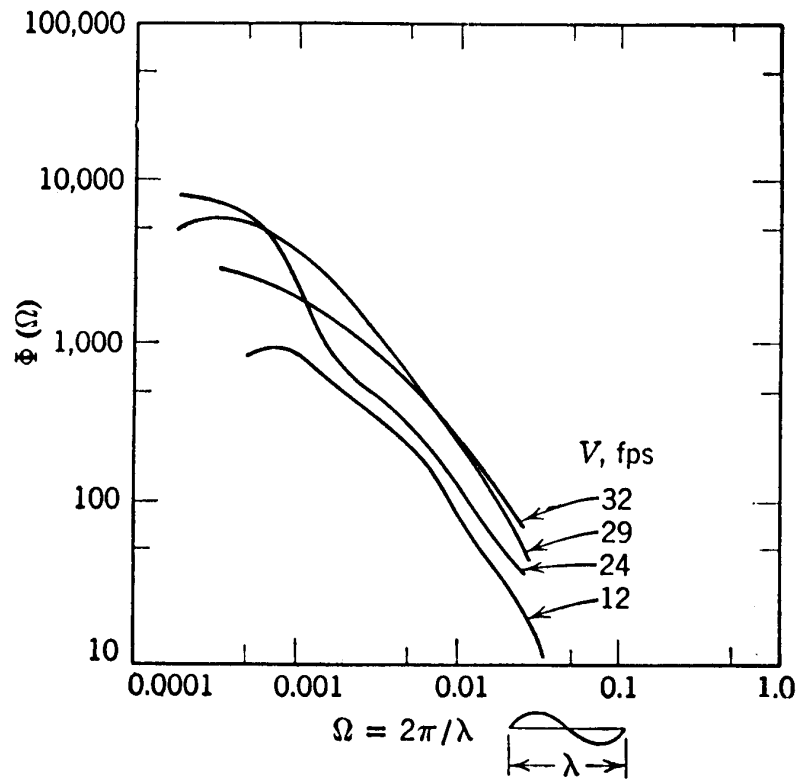


Figure 29. Measurements of the power spectrum of atmospheric turbulence obtained at 300 feet from a meteorological tower [Ref. 16].

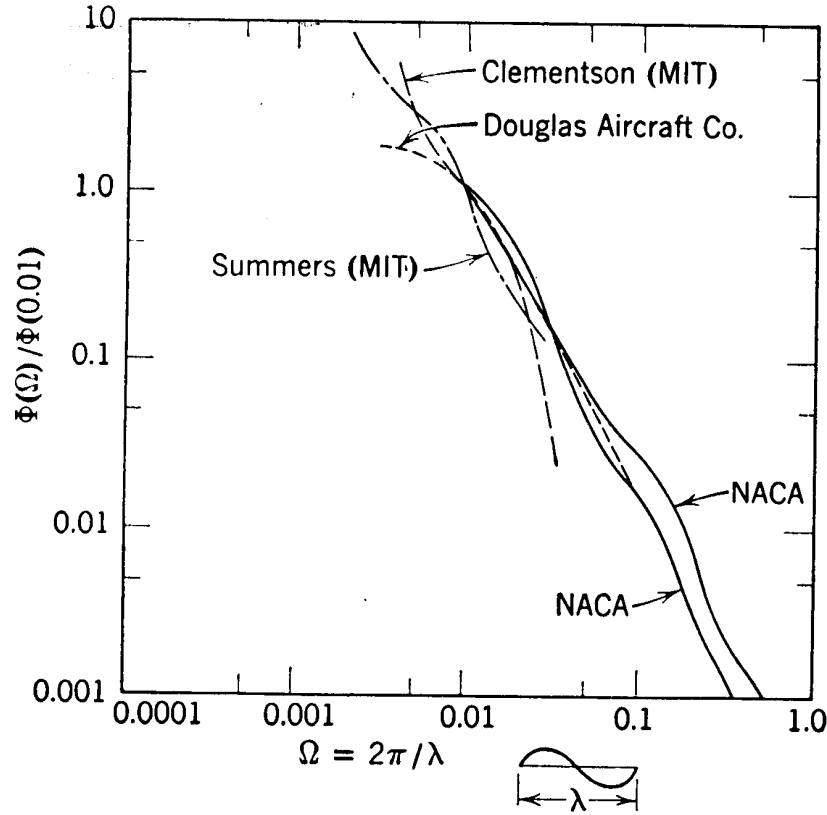


Figure 30. Summary of Airplane Measurements of the Power Spectrum of Atmospheric Turbulence [Ref. 16].

The Dryden model is a function form to empirically fit the power spectral data measured. The Dryden model assumes that the turbulent gusts are random, homogeneous, and isotropic. The assumption of homogeneity implies that the statistical properties of the gusts are the same for each point in the air mass. The assumption of isotropy means that the statistical properties of the air mass do not depend upon spatial orientation. There is a dependence upon the orientation of the airplane because of its motion through the gust field. This follows the background discussion for turbulence models found in [Ref. 3, section 3.7.2]. Taylor's hypothesis assumes that the gust field does not change while the airplane is traversing it because the speed of the airplane is much larger than the speed of the gusts. This hypothesis allows the spatial frequency to be transformed into the time domain by the velocity of the airplane through the field. See [Ref. 5, chapter 10, p 318]. This assumption may be questionable in the case of wind shear, but wind shear has been ignored in this context to simplify gust load analysis.

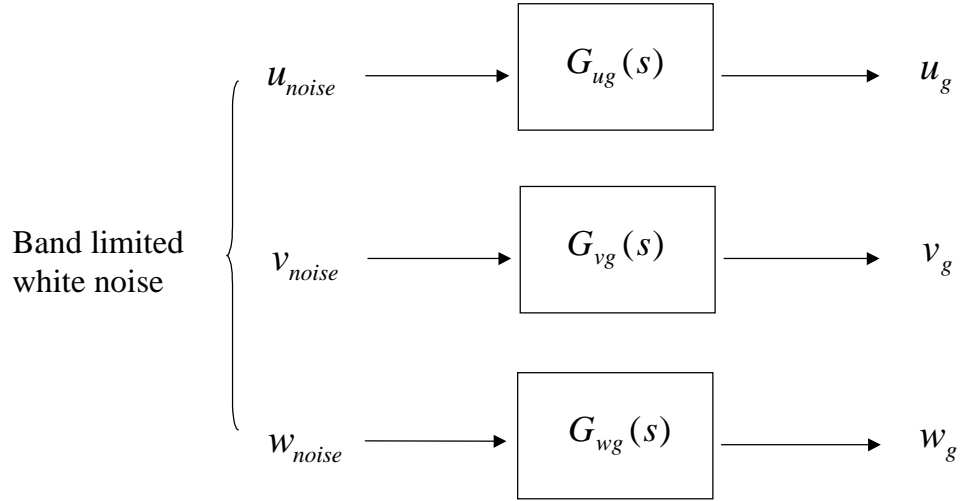


Figure 31. Independent Gaussian Noise Signals Processed by Dryden Shaping Filters to Form Gust Velocity Components

The Dryden gust model uses band limited Gaussian white noise modified by shaping filters to approximate the gust wind velocities in all three components of the body reference frame velocity. The derivation of the gust model from the source data can be described as follows. Wind velocity data is measured, the signal is sampled, and then the autocorrelation function is computed by numerical integration. The power spectral density is calculated by evaluating the Fourier transform using the fast-Fourier-transform (FFT). A polynomial is fitted to the power spectra, which is the Dryden power spectra form. This is outlined in general in [Ref. 7, section 10.6, p 393].

## Dryden Gust Model Notation

Table 5. Dryden Gust Model Notation

$HCG$	Height of the aircraft center of mass (gravity) above ground level. (ft)
$S_u$ $S_v$ $S_w$	Power spectral density
$\Omega$	Spatial frequency (cycles/ft)
$L_u$ $L_v$ $L_w$	Length scale for gust velocity for all three components (ft)  Length scaling for clear air < 1750 ft $L_u = 145(HCG)^{\frac{1}{3}}$ $L_v = 145(HCG)^{\frac{1}{3}}$ $L_w = HCG$
$V_0$	Magnitude of apparent wind velocity from the motion of the aircraft relative to the air mass (inertial velocity adjusted to account for steady wind velocity components)  $V_0 =  \mathbf{V}_{inertial} - \mathbf{V}_{wind} $
$\sigma_u$ $\sigma_v$ $\sigma_w$	Root mean square gust magnitude. (ft/sec)  $\sigma^2 = \int_{-\infty}^{+\infty} S(\Omega) d\Omega$  The U and V components are the same as the input gust magnitude. The W component is scaled by the cube root of the altitude measured from the aircraft center of gravity in feet. This accounts for the boundary condition of zero vertical motion of the gust in contact with the ground. $\sigma_w = \frac{\sigma_u}{\sqrt{145}} (HCG)^{\frac{1}{3}}$

The autocorrelation functions of the Dryden gust model from [Ref. 1, p 42, fig.120] are:

$$\begin{aligned}
 R_{u_g}(\tau) &= e^{-\frac{\tau}{L_u}} \\
 R_{v_g}(\tau) &= e^{-\frac{\tau}{L_v}} \left( 1 - \frac{\tau}{2L_v} \right) \\
 R_{w_g}(\tau) &= e^{-\frac{\tau}{L_w}} \left( 1 - \frac{\tau}{2L_w} \right)
 \end{aligned} \tag{23}$$

The power spectrum functions for the Dryden gust spectra are:

$$\begin{aligned}
 S_{u_g}(\Omega) &= \sigma_u^2 \frac{L_u}{\pi} \frac{1}{1 + (L_u \Omega)^2} \\
 S_{v_g}(\Omega) &= \sigma_v^2 \frac{L_v}{2\pi} \frac{1 + 3(L_v \Omega)^2}{[1 + (L_v \Omega)^2]^2} \\
 S_{w_g}(\Omega) &= \sigma_w^2 \frac{L_w}{2\pi} \frac{1 + 3(L_w \Omega)^2}{[1 + (L_w \Omega)^2]^2}
 \end{aligned} \tag{24}$$

$S_{v_g}(\Omega)$  and  $S_{w_g}(\Omega)$  are of the form of the equation for the lateral one-dimensional spectra given in [Ref. 6, p 318, equation (10.3,12)]. Here is a plot of the v and w component power spectra produced by the Dryden power spectrum functional form.

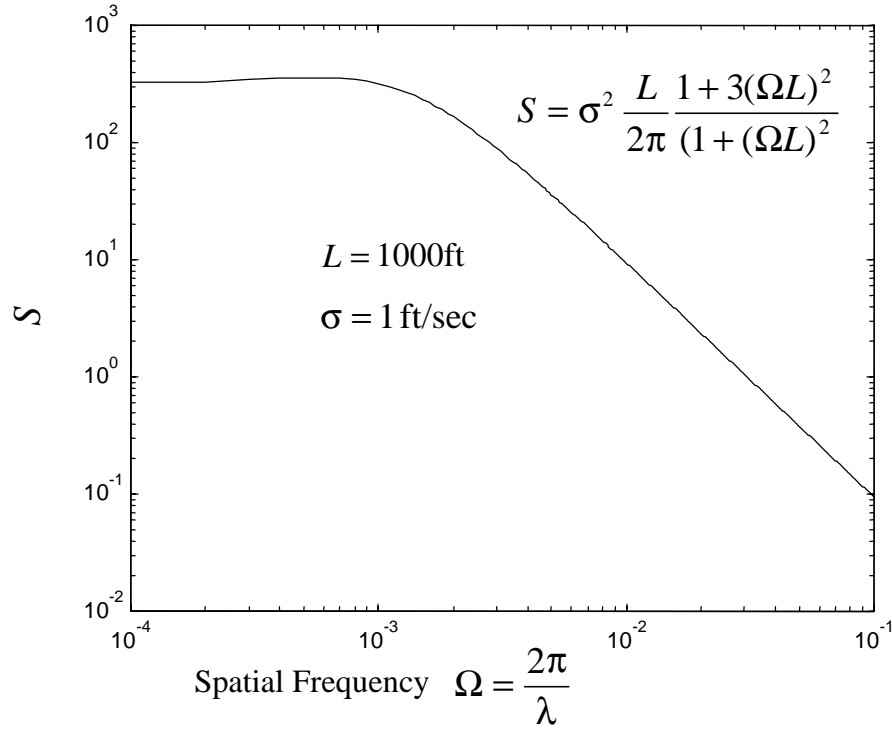


Figure 32. Dryden Spectrum, Components Perpendicular to Flight Direction.

The reference to the “Continuous random model (Dryden form)” given in [Ref. 3, p 424], differs by a factor of two and is wrong. This has been a source of much confusion. The Sperry document [Ref. 9] turns out to use the correct factors though one tends to be fooled by the authoritative appearance of the military specification compared to the less formal Sperry document. An FAA survey document, [Ref. 1, appendix 1C, p 91], confirms this point.

### Laplace Transform Transfer Function Derivation

The process of “spectral factorization” is used to derive the Dryden gust model transfer function for the random number shaping filters. This is described in [Ref. 7, section 10.6, p 393]. The Dryden model is a rational polynomial approximation to the measured wind velocity power spectra. The use of rational polynomials simplifies the design of the Laplace transform transfer functions to shape the white noise for simulated gusts. The use of Gaussian white noise to excite an assumed linear system is a necessary condition to be able to use spectral factorization.

To perform spectral factorization, the power spectrum is first written as a function of  $\omega^2$ .

$$\begin{aligned}
 S_{u_g}(\omega) &= \sigma_u^2 \frac{L_u}{\pi V_0} \frac{1}{1 + \left( \frac{L_u}{V_0} \omega \right)^2} \\
 S_{v_g}(\omega) &= \sigma_v^2 \frac{L_v}{2\pi V_0} \frac{1 + 3 \left( \frac{L_v}{V_0} \omega \right)^2}{\left[ 1 + \left( \frac{L_v}{V_0} \omega \right)^2 \right]^2} \\
 S_{w_g}(\omega) &= \sigma_w^2 \frac{L_w}{2\pi V_0} \frac{1 + 3 \left( \frac{L_w}{V_0} \omega \right)^2}{\left[ 1 + \left( \frac{L_w}{V_0} \omega \right)^2 \right]^2}
 \end{aligned} \tag{25}$$

The notation to describe a linear system undergoing a single impulse input is given here to be consistent with that in [Ref. 7] on spectral factorization. This will be an expanded derivation from that shown in [Ref. 7] applied to the special case of the Dryden gust model. The case of a single scalar input and output modified by a transfer function will be considered as the gust component shaping filter. The function  $H(t)$  is the response of the linear system to an impulse as input. The Laplace transform of the impulse response is the system transfer function in the complex frequency plane  $s$ . The response of the system to random input is also considered. Gaussian white noise is a special form of random input used to simplify the spectral analysis of the system's frequency response. Gaussian white noise has a mean of zero and a flat power spectrum. Use of the white noise abstraction permits simple expressions for the output of the system in the time and frequency domains. The following relations for a linear system come from [Ref. 7, p 390].

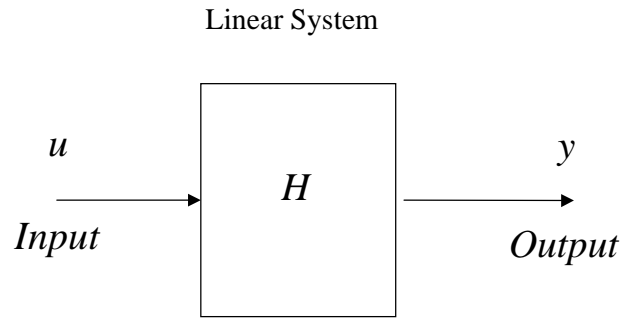


Figure 33. Linear system input-output relationship

Table 6. Laplace Transform Notation

$H(t)$	Impulse response in the time domain
$\mathbf{H}(s)$ , $\mathbf{L}[H(t)]$	Transfer function in the frequency domain

Table 7. Input-output Relation for Linear System with Deterministic Input and with White Noise Input

Domain	Deterministic inputs	White noise inputs
Time domain	$y(t) = \int_0^t H(t - \lambda)u(\lambda)d\lambda$ $\lambda = t - \tau$	$R_y(\tau) = \int_0^\infty H(\xi)QH'(\xi + \tau)d\xi$
Frequency domain	$y(s) = \mathbf{H}(s)u(s)$	$S_y(\omega) = \mathbf{H}(-j\omega)QH'(j\omega)$



## Spectral Factorization

A transfer function expressed in the Laplace transform complex frequency domain  $s$  is the transform of the output function divided by the transform of the input function.

$$\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} \quad (26)$$

[Ref. 7, p 393, 394] shows the process of expressing  $\mathbf{H}(s)$  as the ratio of polynomials in  $s$ , then as a ratio of polynomials in  $\omega^2$ .

$$\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \frac{\beta_1 s^{k-1} + \beta_1 s^{k-2} + \dots + \beta_k}{s^k + \alpha_1 s^{k-1} + \dots + \alpha_k} \quad (27)$$

To perform spectral factorization, reverse the process. Begin with the spectrum expressed as a ratio of polynomials in  $\omega^2$ . Factor the expressions in the numerator and denominator,

$$\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = c \frac{(s - z_1)(s - z_2) \dots (s - z_{k-1})}{(s - p_1)(s - p_2) \dots (s - p_k)} \quad (28)$$

then solve for poles and zeros. Set each factor to zero to find a root of the equation. Roots in the denominator are poles of the transfer function, and roots in the numerator are zeros. Choose the square root for the zero and pole expressions to keep the transfer function in the stable part of the complex plane, the left half-plane. Choose the square root with the positive real part in the numerator and denominator in order to make  $\mathbf{H}(s)$  have poles and zeros in the left half-plane. This is called the “minimum phase” factorization [Ref. 7, p 395].

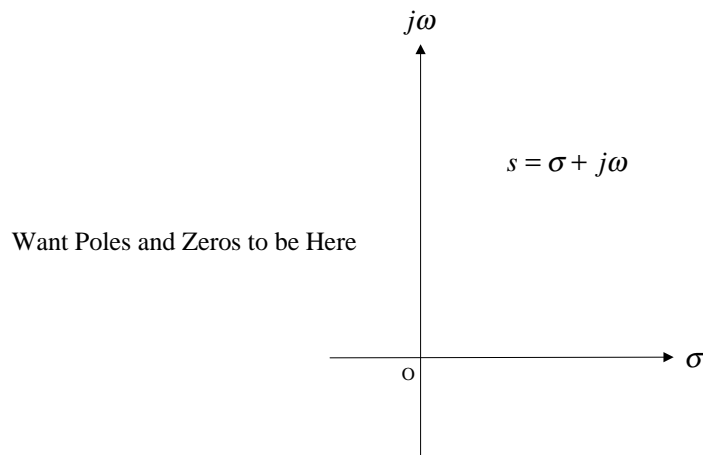


Figure 34. Complex frequency plane

### Step 1.

Write  $S_y$  as a rational function of  $\omega^2$ . To be specific, the transfer function for the  $u_g$  component of the Dryden gust velocity will be calculated. The power spectral density as a rational function of  $\omega^2$  is:

$$S_{u_g}(\omega) = \sigma_u^2 \frac{L_u}{\pi V_0} \frac{1}{1 + \left(\frac{L_u}{V_0}\right)^2 \omega^2} \quad (29)$$

Define the numerator function,

$$N(\omega^2) = 1 \quad (30)$$

and define the denominator function.

$$D(\omega^2) = 1 + \left(\frac{L_u}{V_0}\right)^2 \omega^2 \quad (31)$$

Both are rational functions in  $\omega^2$ .

### Step 2.

Find the roots of  $N(\omega^2)$  and  $D(\omega^2)$

Using the factor theorem from the algebra of nth degree equations from [page 224, Eshbach]:

$$f(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n = 0 \quad (32)$$

*If, and only if,  $(x - r)$  is a factor of  $f(x)$ , then  $f(r) = 0$ .*

First, solve for roots in the numerator.

Numerator:  $N(\omega^2) = 1$  (33)

Notice that the numerator is a constant for all  $\omega$ .

Next, solve for roots in the denominator.

Denominator:  $D(\omega^2) = 1 + \left(\frac{L_u}{V_0}\right)^2 \omega^2$  (34)

$$D(\omega^2) = (\omega^2 - \beta_1)(\omega^2 - \beta_2) \dots (\omega^2 - \beta_k) \quad (35)$$

Taking the first one of the factors,

$$(\omega^2 - \beta_1)$$

To cast the right hand side of equation (34) into this form,

$$\begin{aligned} 0 &= 1 + \left( \frac{L_u}{V_0} \right)^2 \omega^2 \\ 0 &= \left( \frac{L_u}{V_0} \right)^{-2} + \omega^2 \\ 0 &= \omega^2 - (-) \left( \frac{L_u}{V_0} \right)^{-2} \\ \beta &= - \left( \frac{L_u}{V_0} \right)^{-2} \end{aligned} \tag{36}$$

### Step 3.

Find the poles and zeros of the transfer function.

Pole of the transfer function  $p_1 = \sqrt{\beta}$  (37)

$$p_1 = \pm j \left( \frac{L_u}{V_0} \right)^{-1} \tag{38}$$

Make necessary factorization to keep imaginary part on  $\omega$ .

$$\begin{aligned} 0 &= \omega^2 - \beta \\ 0 &= (\omega - \sqrt{\beta})(\omega + \sqrt{\beta}) \\ 0 &= \left[ \omega - j \left( \frac{L_u}{V_0} \right)^{-1} \right] \cdot \left[ \omega + j \left( \frac{L_u}{V_0} \right)^{-1} \right] \\ 0 &= \left[ \left( \frac{L_u}{V_0} \right) j \omega - 1 \right] \cdot \left[ \left( \frac{L_u}{V_0} \right) j \omega + 1 \right] \end{aligned}$$

Substitute the result of the pole and zero calculation back into the power spectral density formula given white noise input.

$$S_{u_g}(\omega) = \mathbf{H}(-j\omega)Q\mathbf{H}'(j\omega) \quad (39)$$

$$S_{u_g}(\omega) = c^2 Q \frac{(\omega^2 - z_1^2) \cdots (\omega^2 - z_{k-1}^2)}{(\omega^2 - p_1^2) \cdots (\omega^2 - p_k^2)} \quad (40)$$

$$S_{u_g}(\omega) = c^2 Q \frac{(-j\omega - z_1) \cdots (-j\omega - z_{k-1})}{(-j\omega - p_1) \cdots (-j\omega - p_k)} \cdot \frac{(j\omega - z_1) \cdots (j\omega - z_{k-1})}{(j\omega - p_1) \cdots (j\omega - p_k)}$$

Make the substitution using the factorized denominator.

$$S_{u_g}(\omega) = c^2 Q \frac{1}{\left[ \left( \frac{L_u}{V_0} \right) j\omega + 1 \right]} \cdot \frac{1}{\left[ \left( \frac{L_u}{V_0} \right) j\omega - 1 \right]} \quad (41)$$

Place the factor in the position for the transfer function  $\mathbf{H}(s)$ . Use the factor that does not have poles in the right half of the complex plane.

#### Step 4.

Choose the transfer function filter gain  $c$ . Let the spectral density of the white noise factor,  $Q$ , be unity for now. It will be accounted for later. Choose  $Q$  to be 1 and  $c^2$  to be  $\sigma_u^2 \frac{L_u}{\pi V_0}$ . The transfer function that results is used to make the longitudinal gust velocity component shaping filter.

$$\mathbf{H}_{u_g}(s) = \sigma_u \sqrt{\frac{L_u}{\pi V_0}} \frac{1}{\left[ \left( \frac{L_u}{V_0} \right) s + 1 \right]} \quad (42)$$

The transfer function for the lateral and vertical gust velocity components can be derived in a similar manner.

$$\mathbf{H}_{v_g}(s) = \sigma_v \sqrt{\frac{L_v}{2\pi V_0}} \frac{1 + \sqrt{3} \frac{L_v}{V_0} s}{\left[ \left( \frac{L_v}{V_0} \right) s + 1 \right]^2} \quad (43)$$

$$\mathbf{H}_{w_g}(s) = \sigma_w \sqrt{\frac{L_w}{2\pi V_0}} \frac{1 + \sqrt{3} \frac{L_w}{V_0} s}{\left[ \left( \frac{L_w}{V_0} \right) s + 1 \right]^2} \quad (44)$$

### Gaussian Band Limited White Noise

This discussion follows one given in the [Ref. 9]. The Dryden model requires Gaussian band-limited white noise as an input to the shaping filters. White noise implies a flat spectrum out to a frequency  $\frac{\omega^*}{2}$  determined by the sampling frame time  $T$ . The spectrum drops to zero at higher frequencies. A coefficient needs to adjust the bandwidth of the transfer functions to match Gaussian white noise. The coefficient  $Q$  will be used to do this.

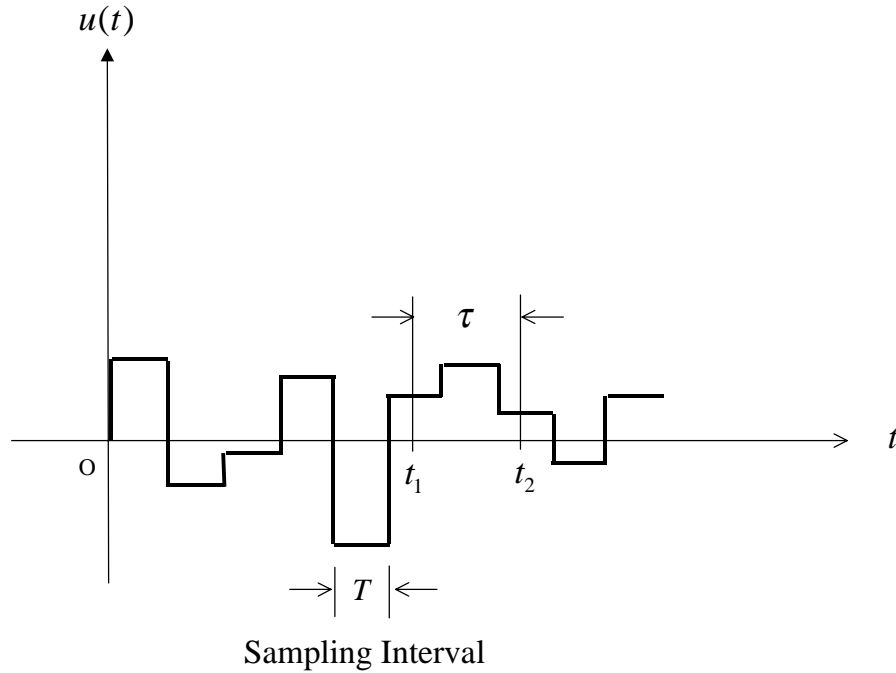


Figure 35. Time history of a turbulence component which approximates a Gaussian distribution.

Gaussian white noise comes from a pseudo-random sequence of numbers which follows a Gaussian probability distribution. The motivation for this comes from the central limit theorem, which states that a Gaussian distribution approximates the distribution of a random variable, which is a function of other random variables, which are independent of each other. The Gaussian distribution has the properties of having a mean of zero and a variance of one. A variance of one also implies a mean square value equal to one.

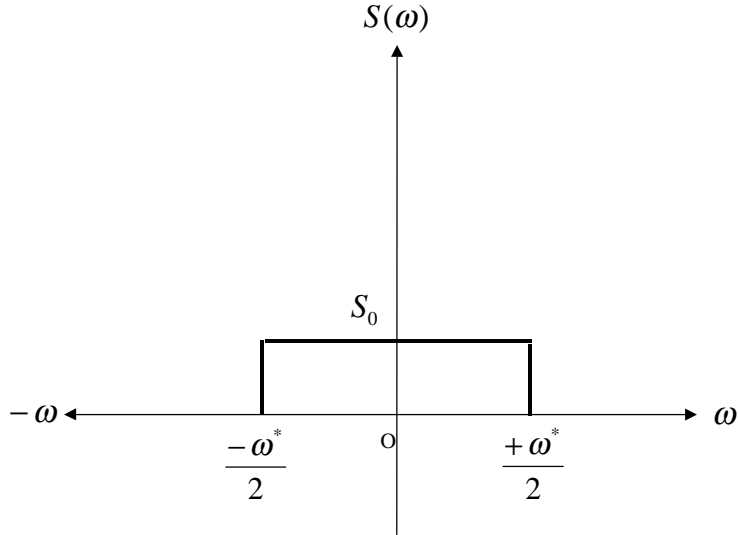


Figure 36. Power spectral density of a turbulence component as approximated by white noise.

The area of the Gaussian probability distribution curve is equal to one. The Gaussian white noise spectrum is assumed to have an area equal to one like the Gaussian distribution. Since the Gaussian white noise spectrum is flat and has an area equal to one, the magnitude  $S_0$  can be solved for in terms of the sampling frequency  $\omega^*$ .

$$S_0 = \frac{1}{\omega^*} = \frac{1}{2\pi f^*} = \frac{T}{2\pi} \quad (45)$$

where

$\omega^*$  is the angular sampling frequency

$f^*$  is the sampling frequency

$T$  is the sampling interval

Following a discussion given in [Ref. 7], adjust the constant  $Q$  in the power spectral density formula to account for Gaussian band limited white noise input.

$$S(\omega) = \mathbf{H}(-j\omega)Q\mathbf{H}'(j\omega) \quad (46)$$

Substitute  $\frac{1}{S_0}$  for  $Q$ .

$$S(\omega) = \mathbf{H}(-j\omega) \frac{1}{S_0} \mathbf{H}'(j\omega)$$

$$S(\omega) = \frac{1}{\sqrt{S_0}} \mathbf{H}(-j\omega) \frac{1}{\sqrt{S_0}} \mathbf{H}'(j\omega)$$

Substitute for the coefficient  $S_0$  from equation (45).

$$S(\omega) = \sqrt{\frac{2\pi}{T}} \mathbf{H}(-j\omega) \sqrt{\frac{2\pi}{T}} \mathbf{H}'(j\omega)$$

Define the coefficients to be part of the transfer function for the shaping filters.

$$S(\omega) = \mathbf{G}(-j\omega) \mathbf{G}'(j\omega) \quad (47)$$

The shaping filter transfer functions with the new coefficients for each gust component are:

$$\mathbf{G}_u(s) = \sigma_u \sqrt{\frac{2L_u}{TV_0}} \frac{1}{\left[ \left( \frac{L_u}{V_0} \right) s + 1 \right]} \quad (48)$$

$$\mathbf{G}_v(s) = \sigma_v \sqrt{\frac{L_v}{TV_0}} \frac{1 + \sqrt{3} \frac{L_v}{V_0} s}{\left[ \left( \frac{L_v}{V_0} \right) s + 1 \right]^2} \quad (49)$$

$$\mathbf{G}_w(s) = \sigma_w \sqrt{\frac{L_w}{TV_0}} \frac{1 + \sqrt{3} \frac{L_w}{V_0} s}{\left[ \left( \frac{L_w}{V_0} \right) s + 1 \right]^2} \quad (50)$$

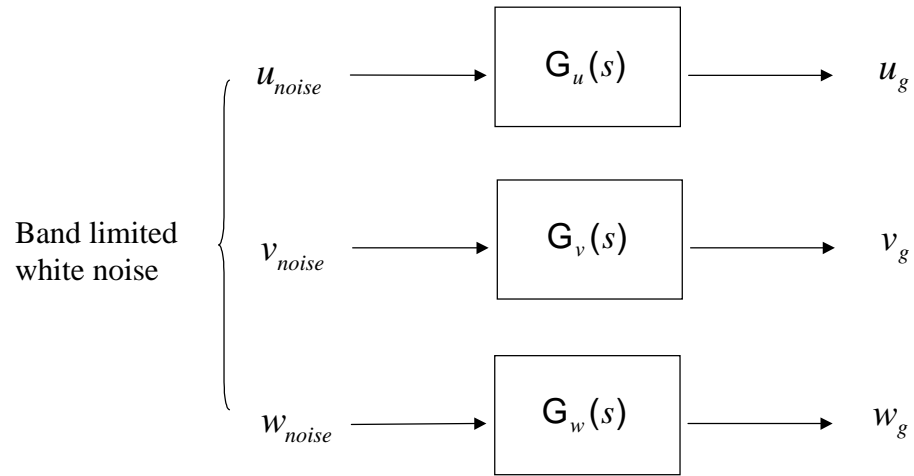


Figure 37. Independent Gaussian noise signals processed by Dryden shaping filters to form gust velocity components.



## Z-transform Transfer Function Derivation

This derivation is expanded from that given in [Ref. 9]. The discrete Z-transform function expression of the Dryden gust model starts off as a modified Laplace transform model. The gust model shaping filter is split into two cascaded filters evaluated in series for the  $v$  and the  $w$  gust components. The first is called the lag filter and the second the lag lead filter.

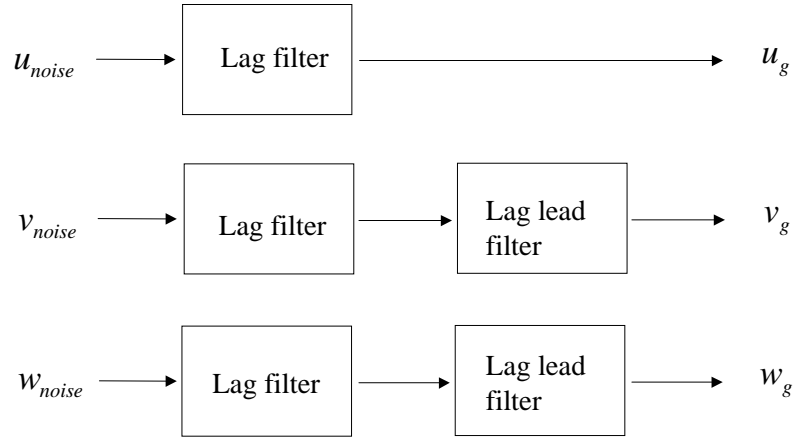


Figure 38. Lag and lag lead filters to replace each shaping filter.

Table 8. Constants for the Series Lag and the Series Lag Lead Filters.

	$u$	$v$	$w$
$\alpha$	$\frac{1}{\sqrt{3}}\beta_u$	$\frac{1}{\sqrt{3}}\beta_v$	$\frac{1}{\sqrt{3}}\beta_w$
$\beta$	$\frac{V_0}{L_u}$	$\frac{V_0}{L_v}$	$\frac{V_0}{L_w}$

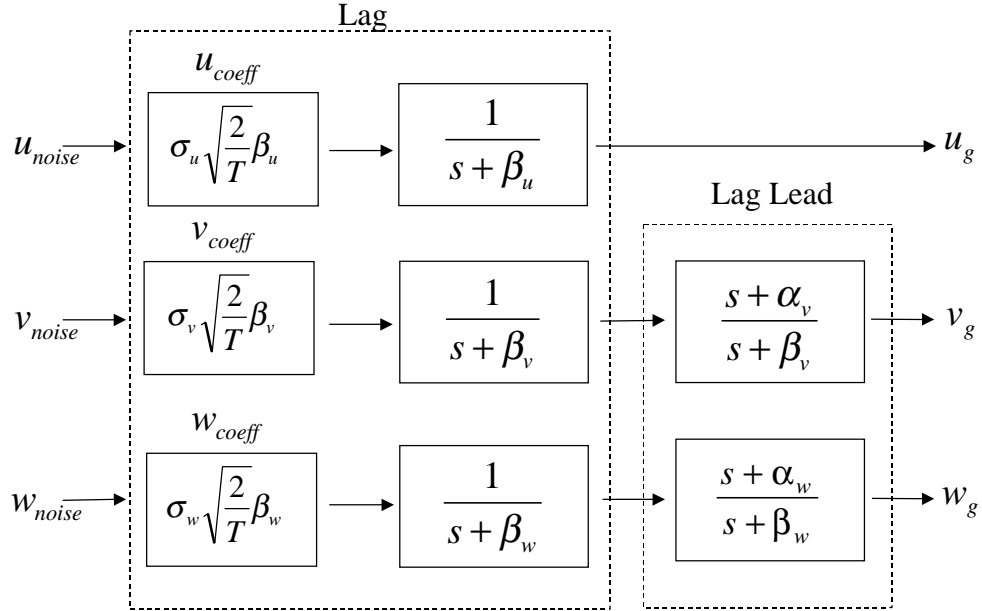


Figure 39. Lag and lead lag filters replace each shaping filter.

The Laplace transform frequency domain expressions for the lag and lead lag filters are found in figure 39. The Z-transform implementation of the continuous lag and lag lead filters follows. First, modify the notation of input and output to a linear system to match that used in [Ref. 15].

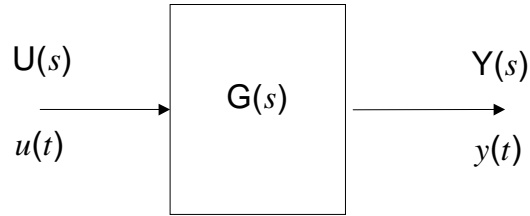


Figure 40. Linear system with transfer function  $\mathbf{G}(s)$ .

To convert from the Gaussian white noise input to the gust component output, a transfer function represented by  $\mathbf{G}(s)$  is used. This was derived from the Dryden theory of wind gusts through the process of spectral factorization.

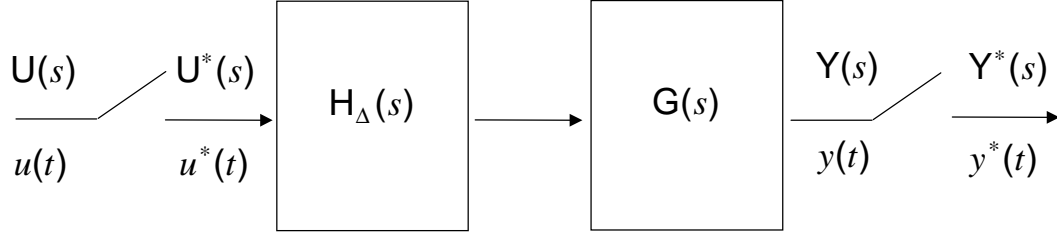


Figure 41. Sampled equivalent of a linear system with transfer function  $\mathbf{G}(s)$ .

Figure 41 shows the linear system with the triangular hold transfer function added in series. The linear system has a continuous function input  $u(t)$  that is represented by the graph in figure 42(a). The asterisk notation means “a sample of”. The switch represents a digital sampler. An equivalent ideal sampler can be imagined to exist and is represented mathematically by a train of delta functions of unit amplitude occurring at  $T$  instants in time. See figure 42(b). The same thing shown symbolically is:

$$\delta_T(t) = \delta(t-0) + \delta(t-T) + \delta(t-2T) + \delta(t-3T) + \dots + \delta(t-kT)$$

$$\delta_T(t) = \sum_{k=0}^{\infty} \delta(t-kT) \quad (51)$$

The function  $u(t)$  when sampled at the  $T$  instants in time can be thought of as being the product of the value of the continuous function and the delta function. This new function is given the name  $u^*(t)$  using the asterisk notation. This train of scaled impulses is shown in figure 42(c).  $u^*(t)$  shown symbolically is:

$$u^*(t) = u(0)\delta(t-0) + u(T)\delta(t-T) + u(2T)\delta(t-2T) + \dots + u(kT)\delta(t-kT)$$

Which is an infinite series expressed as:

$$u^*(t) = \sum_{k=0}^{\infty} u(kT)\delta(t-kT) \quad (52)$$

To solve the differential control equations algebraically, the Laplace transform representation is useful. It transforms the equations in time to be equations in the complex frequency plane  $s$ . The definition of the Laplace transform is:

$$\mathbf{L}[u(t)] = \mathbf{U}(s) = \int e^{-ts} u(t) dt \quad (53)$$

The Laplace transform of  $u^*(t)$  is given by:

$$\mathbf{U}^*(s) = u(0)\mathbf{L}[\delta(t-0)] + u(T)\mathbf{L}[\delta(t-T)] + u(2T)\mathbf{L}[\delta(t-2T)] + \dots + u(kT)\mathbf{L}[\delta(t-kT)]$$

When the function is sampled, the terms only exist for  $t = 0, T, 2T, \dots, kT$ .

$$\mathbf{U}^*(s) = u(0)e^0 + u(T)e^{-Ts} + u(2T)e^{-2Ts} + \dots + u(kT)e^{-kTs}$$

The result is the Laplace transform of a sampled function expressed as an infinite series.

$$\mathbf{U}^*(s) = \sum_{k=0}^{\infty} u(kT)e^{-kTs} \quad (54)$$

At this point, define  $z = e^{Ts}$  and substitute to get the z transform that corresponds to the Laplace transform sampled periodically.

$$\mathbf{U}(z) = \mathbf{U}^*(s) \Big|_{s=\frac{1}{T}\ln z} = \sum_{k=0}^{\infty} u(kT)z^{-k} \quad (55)$$

Notice that  $s$  transforms into  $z$  with

$$s = \frac{1}{T} \ln z. \quad (56)$$

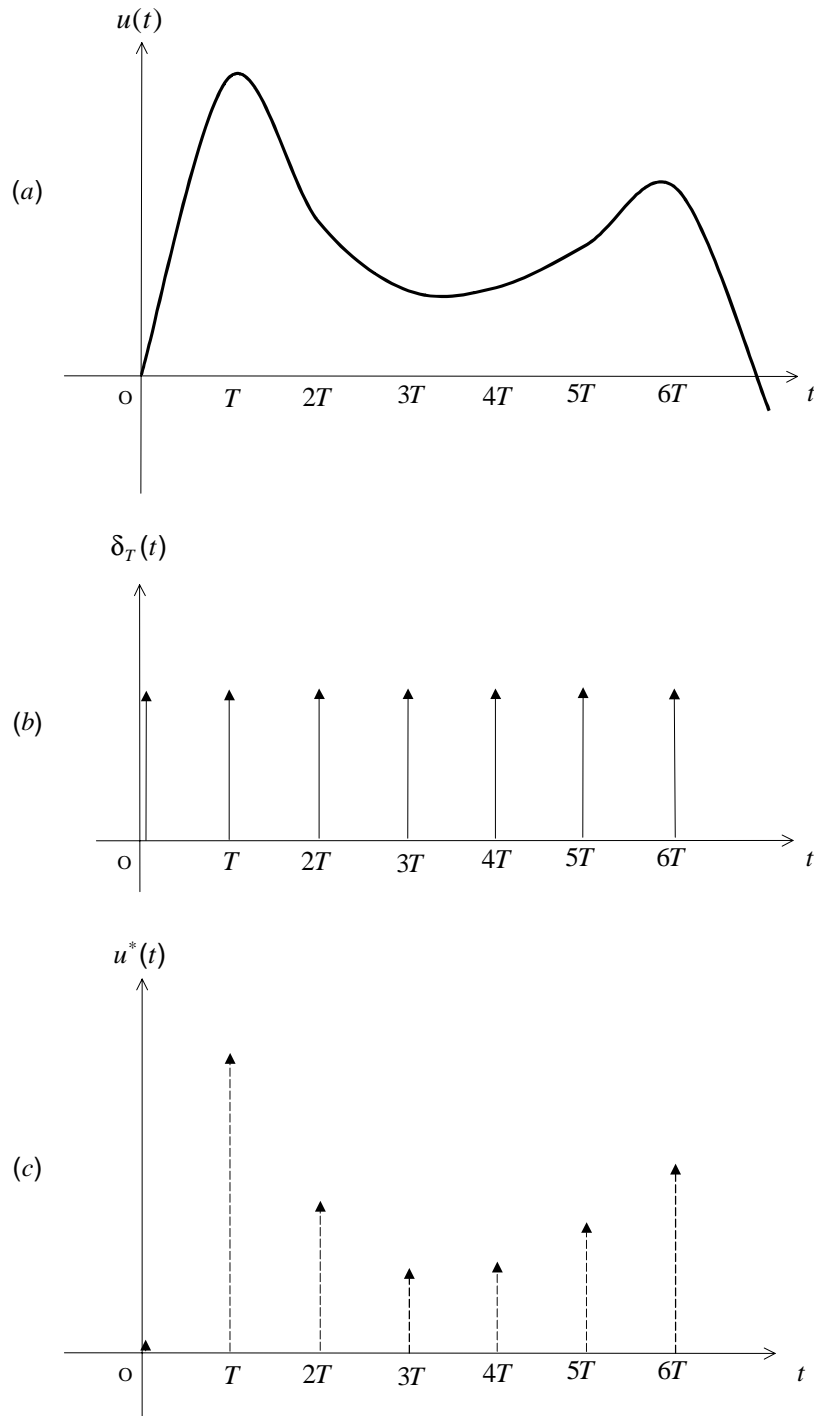


Figure 42. (a) A continuous function. (b) A train of impulse functions. (c) A train of impulse functions multiplied by the value of the continuous function at each sample instant

The ideal sampler, represented in figure 41 by a switch, is used to get samples of  $U(s)$  at  $T$  instants in time. This time history of impulses or “spikes” have high frequency components and will affect the filter  $G(s)$  very differently than the original continuous signal. To prevent this distortion, a reconstruction is made using a polygonal hold block. The polygonal (also called triangular hold) reconstructs the signal and has lower frequency components and much more closely matches the original signal being applied to  $G(s)$ . Figure 43 shows how a continuous function  $u(t)$  can be approximated using this polygonal hold technique [Ref. 18].

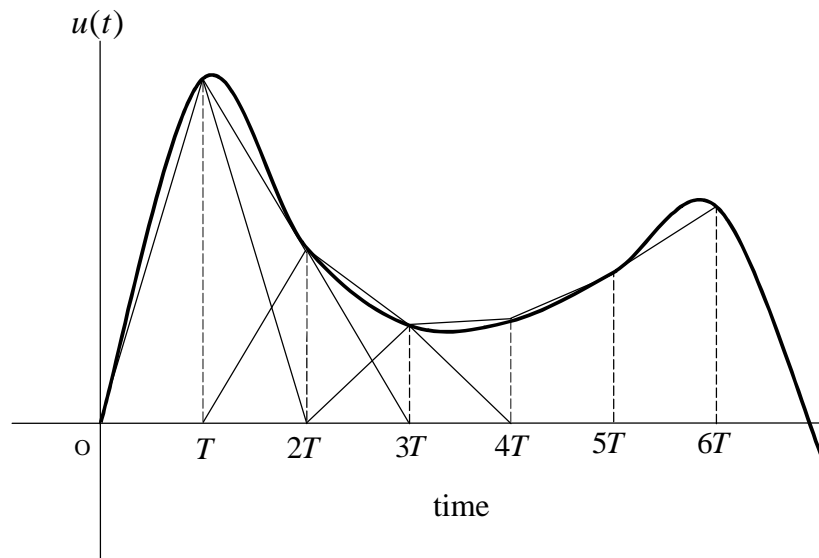


Figure 43. A polygonal hold piecewise linear approximation to the continuous function  $u(t)$ .

The polygonal hold is like the trapezoidal rule of numerical integration. The function being approximated is replaced by piecewise linear chord segments of the curve. The chord slopes can be thought of as being constructed from the sum of pairs of triangular functions.

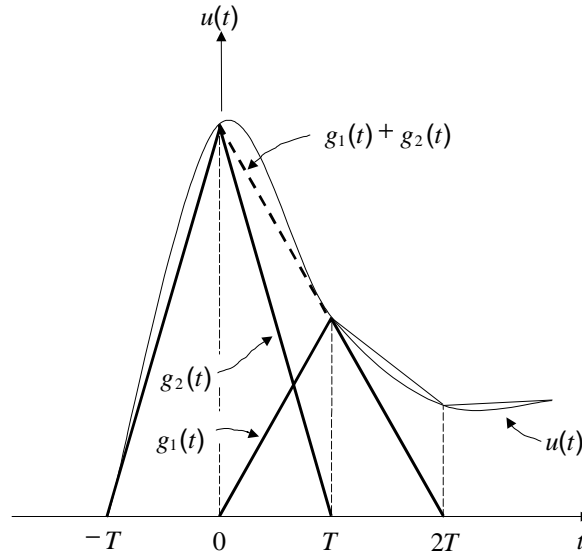


Figure 44. Details of the two triangular generating functions used to generate a segment of the polygonal hold approximation.

$$g_1(t) = 0 + t \frac{u(T)}{T}$$

$$g_2(t) = u(0) + t \left( 0 - \frac{u(0)}{T} \right)$$

$$g_1(t) + g_2(t) = u(0) + t \left( \frac{u(T) - u(0)}{T} \right)$$

This mathematical construction requires a starting point  $-T$  and temporary origin at each sample point. This triangle can be further thought of as being constructed by the superposition of three ramp functions. These ramp functions can be converted into their Laplace transform equivalents more easily than the original construction. This solution is a graphical aid to justify the process. Another method uses a set of infinite series to make this approximation in a more formal mathematical way but is not included here.

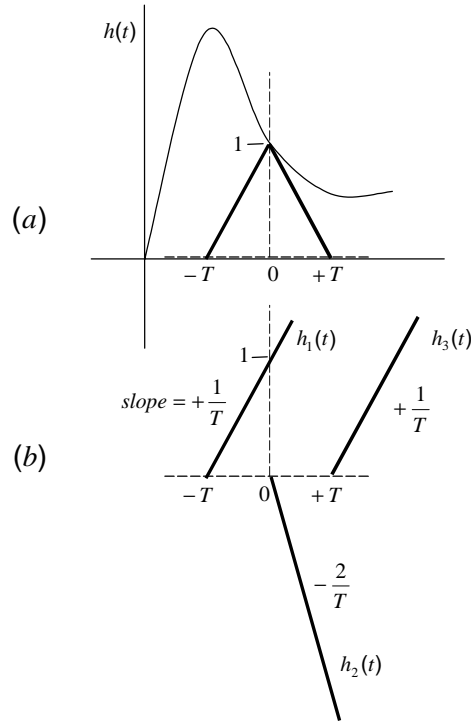


Figure 45. The triangular hold function as the superposition of simple ramp functions [Ref. 18].

The triangle generating function as a sum of ramp equations is:

$$\begin{aligned}
 h_1(t) &= \frac{1}{T}(t + T) \\
 h_2(t) &= -\frac{2}{T}t \\
 h_3(t) &= \frac{1}{T}(t - T)
 \end{aligned}
 \tag{57}$$

The Laplace transform of each ramp equation according to the Laplace transform definition is

$$\begin{aligned}
 \mathbf{H}_1(s) &= \frac{1}{T} \int_{-T}^{\infty} t e^{-st} dt + \int_{-T}^{\infty} e^{-st} dt \\
 \mathbf{H}_2(s) &= -\frac{2}{T} \int_0^{\infty} t e^{-st} dt
 \end{aligned}
 \tag{58}$$



$$\mathbf{H}_3(s) = \frac{1}{T} \int_T^{\infty} t e^{-st} dt - \int_T^{\infty} e^{-st} dt .$$

Begin the integration at one time step before time zero to permit this construction. This amounts to a modification to the usual definition of integration limits of the Laplace transform. The function being transformed can be restricted to be zero during the one negative step to prevent any convergence problems. It can also be looked at as an approximation that neglects only one step and will be very similar to the result if the integration limit started at zero. Change the integration limit to allow one integral term to be split in two. This technique is used to make the integration limits of several terms match so that they can be algebraically added and canceled.

$$\begin{aligned} \mathbf{H}_1(t) &= \frac{1}{T} \int_T^{\infty} t e^{-st} dt + \frac{1}{T} \int_{-T}^T e^{-st} dt + \int_T^{\infty} e^{-st} dt \\ \mathbf{H}_2(t) &= -\frac{2}{T} \int_T^{\infty} t e^{-st} dt - \frac{2}{T} \int_0^T t e^{-st} dt \end{aligned} \quad (59)$$

$$\mathbf{H}_3(t) = \frac{1}{T} \int_T^{\infty} t e^{-st} dt - \int_T^{\infty} e^{-st} dt$$

Add the Laplace transform of each of the three ramp functions together.

$$\mathbf{H}_{\Delta}(s) = \mathbf{H}_1(s) + \mathbf{H}_2(s) + \mathbf{H}_3(s)$$

The first term of each of the component functions when added cancel to zero, and the last term of  $\mathbf{H}_1(s)$  and  $\mathbf{H}_3(s)$  cancel. The remaining terms sum to

$$\mathbf{H}_{\Delta}(s) = \frac{1}{T} \int_{-T}^T t e^{-st} dt + \int_{-T}^T e^{-st} dt - \frac{2}{T} \int_0^T t e^{-st} dt . \quad (60)$$

Use integration by parts on the first and last term.

$$\mathbf{H}_{\Delta}(s) = -\frac{1}{T} \left[ t \frac{e^{-st}}{s} + \frac{e^{-st}}{s^2} \right]_{-T}^T - \left[ \frac{e^{-st}}{s} \right]_{-T}^T + \frac{2}{T} \left[ t \frac{e^{-st}}{s} + \frac{e^{-st}}{s^2} \right]_0^T$$

Factor the result to get the expression in [Ref. 18].

$$\mathbf{H}_{\Delta}(s) = \frac{1}{Ts^2} e^{Ts} - \frac{2}{Ts^2} + \frac{1}{Ts^2} e^{-Ts} . \quad (61)$$

Multiply this expression by  $e^{Ts} e^{-Ts}$  and regroup to factor and obtain the expression for the triangular hold transfer function in [Ref. 9].

$$\mathbf{H}_{\Delta}(s) = \frac{e^{+Ts}(1 - e^{-Ts})^2}{Ts^2} \quad (62)$$

The digitally sampled equivalent of  $\mathbf{G}(s)$  is  $\mathbf{G}(z)$ .

$$\mathbf{G}(z) = \mathbf{Z}[\mathbf{H}_{\Delta}(s)\mathbf{G}(s)] \quad (63)$$

Likewise  $\mathbf{U}(s)$  is replaced by  $\mathbf{U}(z)$ , and  $\mathbf{Y}(s)$  by  $\mathbf{Y}(z)$ .

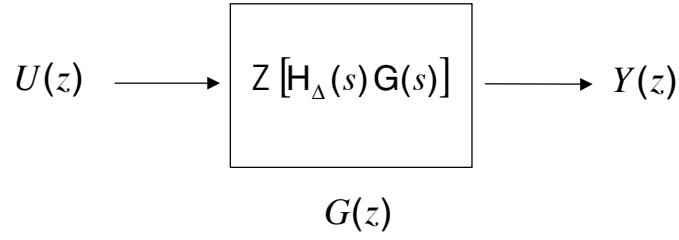


Figure 46. Z-transform equivalent of a continuous linear system.

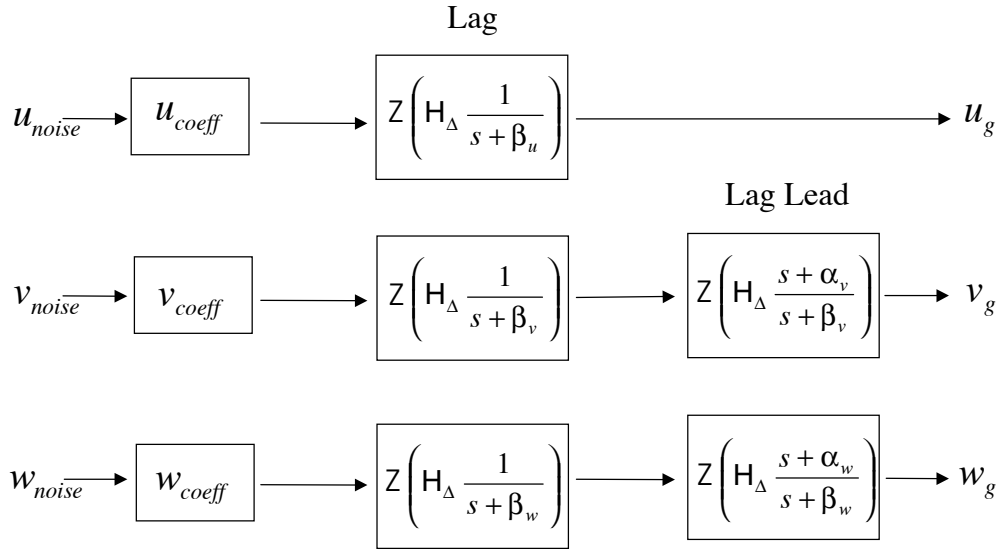


Figure 47. Z-transform version of the shaping filters showing the component Laplace transform expressions and the sampling function.

### Lag Filter Case

The conversion of the Laplace lag filters to the Z-transform representation is described here. First, write the lag block in terms of  $s$ . Then insert the sample and hold. The triangular hold transfer function and the lag transfer function are converted to the Z-transform representation together. This section closely follows an example from [Ref. 15, p 165-166]. Start with the combined triangular hold and lag filter transfer functions. This analysis applies to all three Cartesian velocity components  $u, v, w$ . Leave the velocity subscripts off for clarity.

$$\frac{\mathbf{Y}_{lag}(s)}{\mathbf{U}_{lag}(s)} = \frac{e^{Ts}(1 - e^{-Ts})^2}{Ts^2} \frac{1}{s + \beta} \quad (64)$$

Let  $\mathbf{G}(s)$  represent  $\frac{e^{Ts}(1 - e^{-Ts})}{Ts} \frac{1}{s + \beta}$  to be consistent with the notation used in [Ref. 15]. The product of the remaining  $\frac{(1 - e^{-Ts})}{s}$  factor with the rest becomes

$$\mathbf{X}(s) = \frac{(1 - e^{-Ts})}{s} \mathbf{G}(s) \quad (65)$$

To find the Z-transform of this product, move the  $s$  in the denominator to be under  $\mathbf{G}(s)$  and define  $\mathbf{G}_1(s)$  to contain the  $s$

$$\mathbf{X}(s) = (1 - e^{-Ts}) \frac{\mathbf{G}(s)}{s} = (1 - e^{-Ts}) \mathbf{G}_1(s) \quad (66)$$

$$\text{where } \mathbf{G}_1(s) = \frac{\mathbf{G}(s)}{s}.$$

Use the distributive law to multiply out the product and examine the second term.

$$\mathbf{X}_1(s) = e^{-Ts} \mathbf{G}_1(s) \quad (67)$$

The function  $\mathbf{X}_1(s)$  can be considered the product of two Laplace transformed functions  $e^{-Ts}$  and  $\mathbf{G}_1(s)$ . The product of any pair of Laplace transformed functions in the  $s$ -domain corresponds to the following convolution integral in the  $t$ -domain from the basic properties of the Laplace transform [Ref. 10, p 18].

$$x_1(t) = \int_0^t g_0(t - \tau) g_1(\tau) d\tau \quad (68)$$

Let:

$$g_0(t) = \mathbf{L}^{-1}[e^{-Ts}]$$

$$g_1(t) = \mathbf{L}^{-1}[\mathbf{G}_1(s)]$$

Next, perform a side calculation to get  $g_0(t)$ .

To find  $g_0(t)$ , use the delta function definition and the Laplace transform definition. The delta function (impulse function) is a special limiting process that returns a one for an argument of zero and zero everywhere else. From the integral definition of the Laplace transform obtain:

$$\mathbf{L}[g_0(t)] = \int_0^{\infty} e^{-ts} g_0(t) dt \quad (69)$$

Since the Laplace transform looks like the exponential factor of the integrand, try using the property of the delta function that pulls whatever is multiplying it outside the integral sign when its argument is zero.

Try  $\delta(t - kT)$  as  $g_0(t)$ .

$$\mathbf{L}[g_0(t)] = \int_0^{\infty} e^{-ts} \delta(t - kT) dt \quad (70)$$

$\delta(t - kT)$  is non zero only for  $t = kT$ . It pulls the function value out of the integral only for values of  $t = kT$ .

$$\mathbf{L}[g_0(t)] = \mathbf{L}[\delta(t - kT)] = e^{-kTs}$$

For this case  $k = 1$ .

$g_0(t) = \delta(t - T)$  is the time domain function corresponding to  $e^{-Ts}$ .

Now, returning to the convolution integral calculation, the two factors of the integrand are given by

$$g_0(t) = \delta(t - T)$$

$$g_1(t) = \mathbf{L}^{-1}[\mathbf{G}_1(s)].$$

The delta function selects the value of  $g_1(t)$  at time  $(t - T)$  when  $\tau$  is  $(t - T)$ .

$$x_1(t) = \int_0^t \delta(t - T - \tau) g_1(\tau) d\tau$$

$$x_1(t) = g_1(t - T)$$

Take the Z-transform of both sides:

$$\mathbf{Z}[x_1(t)] = \mathbf{Z}[g_1(t - T)]$$

Define the Z-transform of the function  $g_1(t)$  to be

$$\mathbf{G}_1(z) = \mathbf{Z}[g_1(t)]. \quad (71)$$

From the Z-transform real translation theorem (shifting theorem), [Ref. 15, p 52], the Z-transform of  $g_1(t)$  is

$$\mathbf{X}_1(z) = \mathbf{Z}[x_1(t)] = \mathbf{Z}[g_1(t - T)] = z^{-1} \mathbf{G}_1(z). \quad (72)$$

From equation (66),

$$\mathbf{X}(s) = (1 - e^{-Ts}) \mathbf{G}_1(s)$$

taking the Z-transform of both sides we get

$$\mathbf{Z}[\mathbf{X}(s)] = \mathbf{Z}[(1 - e^{-Ts}) \mathbf{G}_1(s)]$$

$$\mathbf{X}(z) = \mathbf{Z}[\mathbf{G}_1(s) - e^{-Ts} \mathbf{G}_1(s)].$$

Replace the Z-transform from the  $s$ -domain of the first term with the corresponding Z-transform from the  $t$ -domain.

$$\mathbf{X}(z) = \mathbf{Z}[g_1(t)] - \mathbf{Z}[e^{-Ts} \mathbf{G}_1(s)] \quad (73)$$

Use equation (67) to substitute for the second term.

$$\mathbf{X}(z) = \mathbf{Z}[g_1(t)] - \mathbf{Z}[\mathbf{X}_1(s)]$$

$$\mathbf{X}(z) = \mathbf{G}_1(z) - \mathbf{X}_1(z) \quad (74)$$

Use equation (72) to substitute in place of the second term of equation (74) and factor common terms.

$$\mathbf{X}(z) = \mathbf{G}_1(z) - z^{-1}\mathbf{G}_1(z)$$

$$\mathbf{X}(z) = (1 - z^{-1})\mathbf{G}_1(z) \quad (75)$$

To summarize this process, if  $\mathbf{X}(s)$  involves a factor  $(1 - e^{-Ts})$ , then to obtain the Z-transform of  $\mathbf{X}(s)$ , the term  $(1 - e^{-Ts})$  may be factored out so that  $\mathbf{X}(z)$  becomes the product of  $(1 - z^{-1})$  and the Z-transform of the remaining term. As a check, remember that from the definition of  $z$  from sampled control theory,  $z = e^{sT}$  [Ref. 15, p 150]. Make this substitution to verify the correspondence between  $(1 - e^{-Ts})$  and  $(1 - z^{-1})$ .

$$\mathbf{X}(z) = \mathbf{Z}[\mathbf{X}(s)] = (1 - z^{-1})\mathbf{Z}\left[\frac{\mathbf{G}(s)}{s}\right] \quad (76)$$

Substitute back in for  $\frac{\mathbf{G}(s)}{s}$  and repeat the process to get the Z-transform of the second  $(1 - e^{-Ts})$  factor. Take the Z-transform of the second  $(1 - e^{-Ts})$  factor and the rest:

$$(1 - z^{-1})\mathbf{Z}\left[e^{Ts}(1 - e^{-Ts})\frac{1}{Ts^2(s + \beta)}\right]$$

In like manner to the process that pulled out the  $(1 - z^{-1})$  factor, the  $(1 - e^{-Ts})$  factor is pulled out of the Z-transform operator.

$$\mathbf{X}(z) = (1 - z^{-1})(1 - z^{-1})\mathbf{Z}\left[e^{Ts}\frac{1}{Ts^2(s + \beta)}\right] \quad (77)$$

Consider the remaining Z-transformed product:

$$\mathbf{Z}\left[e^{Ts}\frac{1}{Ts^2(s + \beta)}\right]$$

Again, let  $\mathbf{G}(s)$  be redefined to represent  $\frac{1}{Ts(s + \beta)}$  with  $\mathbf{G}_1(s) = \frac{\mathbf{G}(s)}{s}$ .

Redefine  $\mathbf{X}_1(s)$  to be the product of the two Laplace transformed functions  $e^{Ts}$  and  $\mathbf{G}_1(s)$ .

$$\mathbf{X}_1(s) = e^{Ts}\mathbf{G}_1(s) \quad (78)$$

Again, use the Laplace transform convolution integral of a product of Laplace transformed functions to get the output function of a linear time-invariant system.

$$x_1(t) = \int_0^t g_0(t-\tau)g_1(\tau)d\tau \quad (79)$$

Let:

$$\mathbf{G}_0(s) = e^{Ts}$$

$$\mathbf{G}_1(s) = \frac{1}{Ts^2(s + \beta)}$$

$$g_0(t) = \mathbf{L}^{-1}[e^{Ts}]$$

$$g_1(t) = \mathbf{L}^{-1}[\mathbf{G}_1(s)]$$

Next, perform a side calculation to get  $g_0(t)$ .

As before, use the definition of the Laplace transform and the delta function considered as a possible solution.

Since the Laplace transform looks like the exponential factor of the integrand, try using the property of the delta function that pulls whatever is multiplying it outside the integral sign when its argument is zero. This time there is a positive exponential result desired. So the delta function is advanced by  $T$  instead of delayed by  $kT$ .

Try  $\delta(t + T)$  as  $g_0(t)$ .

$$\mathbf{L}[g_0(t)] = \int_0^{\infty} e^{-ts} \delta(t + T) dt \quad (80)$$

$\delta(t + T)$  is non zero only for  $t = -T$ . It pulls the function value out of the integral only for values of  $t = -T$ . This is a problem since it is just to the negative of the limits of integration. The limits of integration will be extended on sample period  $T$  before zero. This approximation depends upon the function being small or zero when time is less than zero.

$$\mathbf{L}[g_0(t)] = \int_{-T}^{\infty} e^{-ts} \delta(t+T) dt \quad (81)$$

$$\mathbf{L}[g_0(t)] = e^{Ts}$$

$$g_0(t) = \delta(t+T)$$

$g_0(t) = \delta(t+T)$  is the time domain function corresponding to  $e^{Ts}$ .

Now, back to the calculation of the convolution integral where the two factors of the integrand are given by

$$g_0(t) = \mathbf{L}^{-1}[e^{Ts}] = \delta(t+T)$$

$$g_1(t) = \mathbf{L}^{-1}[\mathbf{G}_1(s)]$$

Use the delta function to select the value of  $g_1(t)$  at time  $\tau = -T$ . To make time  $-T$  under the integral, the limits of integration must be extended one sample time  $T$  before zero. This approximation is like the earlier one made with the Laplace transform of the exponential function derivation.

$$x_1(t) = \int_{-T}^t g_0(t-\tau) g_1(\tau) d\tau \quad (82)$$

Substitute for  $g_0(t)$ .

$$x_1(t) = \int_{-T}^t \delta(t+T-\tau) g_1(\tau) d\tau$$

The integral is non-zero only for  $\tau = t+T$ .

$$x_1(t) = g_1(t+T)$$

Take the Z-transform of both sides:

$$\mathbf{Z}[x_1(t)] = \mathbf{Z}[g_1(t+T)]$$

From the Z-transform real translation theorem (shifting theorem), [Ref. 15, p 52], the Z-transform of  $g_1(t)$  is

$$\mathbf{X}_1(z) = \mathbf{Z}[x_1(t)] = \mathbf{Z}[g_1(t+T)] = z\mathbf{G}_1(z) - zx_1(0). \quad (83)$$



The second term can be neglected for  $x_1(0)$  small or zero at time zero.

$$\mathbf{X}_1(z) \cong z\mathbf{G}_1(z) \quad (84)$$

Substitute back into equation (77).

$$\mathbf{X}(z) \cong z(1 - z^{-1})^2 \mathbf{Z} \left[ \frac{1}{Ts^2(s + \beta)} \right] \quad (85)$$

In the table of Z-transforms [Ref. 15, p50], there is an entry of the form

$$\frac{a^2}{s^2(s + a)}, \text{ corresponding to } \frac{[(aT - 1 + e^{-aT}) + (1 - e^{-aT} - aTe^{-aT})z^{-1}]z^{-1}}{(1 - z^{-1})^2(1 - e^{-aT}z^{-1})}. \quad (86)$$

Replace  $a$  by  $\beta$  and divide the right side of equation (85) by  $\beta^2$  to match the factor to the one from the table.

$$z(1 - z^{-1})^2 \frac{1}{T\beta^2} \mathbf{Z} \left[ \frac{\beta^2}{s^2(s + \beta)} \right] \quad (87)$$

Plug in the Z-transform from the table and obtain:

$$z(1 - z^{-1})^2 \frac{1}{T\beta^2} \frac{[(\beta T - 1 + e^{-\beta T}) + (1 - e^{-\beta T} - \beta Te^{-\beta T})z^{-1}]z^{-1}}{(1 - z^{-1})^2(1 - e^{-\beta T}z^{-1})}$$

Simplify and rearrange

$$\frac{1}{T\beta^2} \frac{(\beta T - 1 + e^{-\beta T}) + (1 - e^{-\beta T} - \beta Te^{-\beta T})z^{-1}}{(1 - e^{-\beta T}z^{-1})} \quad (88)$$

The following is taken from the definition of a transfer function from linear theory:

$$\frac{\text{Output}(z)}{\text{Input}(z)} = \text{the transfer function}$$

$$\frac{\mathbf{Y}(z)}{\mathbf{U}(z)} = \frac{1}{T\beta^2} \frac{(\beta T - 1 + e^{-\beta T}) + (1 - e^{-\beta T} - \beta Te^{-\beta T})z^{-1}}{(1 - e^{-\beta T}z^{-1})}$$

Arrange terms using algebra:

$$\begin{aligned}\mathbf{Y}(z)(1 - e^{-\beta T} z^{-1}) &= \frac{1}{T\beta^2} (\beta T - 1 + e^{-\beta T}) \mathbf{U}(z) + \frac{1}{T\beta^2} (1 - e^{-\beta T} - \beta T e^{-\beta T}) z^{-1} \mathbf{U}(z) \\ \mathbf{Y}(z) &= e^{-\beta T} z^{-1} \mathbf{Y}(z) + \frac{1}{T\beta^2} (1 - e^{-\beta T} - \beta T e^{-\beta T}) z^{-1} \mathbf{U}(z) + \frac{1}{T\beta^2} (\beta T - 1 + e^{-\beta T}) \mathbf{U}(z)\end{aligned}\quad (89)$$

The real translation theorem (shifting theorem) of the Z-transform from [Ref. 15, p 52]:

If  $x(t) = 0$  for  $t < 0$  and  $x(t)$  has the  $z$  transform  $\mathbf{X}(z)$ , then  $x(t - nT) = z^{-n} \mathbf{X}(z)$

where  $n$  is zero or a positive integer.

For the case where  $n$  is 0,

$$x(t) = \mathbf{X}(z) .$$

For the case where  $n$  is 1,

$$x(t - T) = z^{-1} \mathbf{X}(z) .$$

Consider only the instants of time when sample frames are taken  $kT$  where  $k$  is the frame number:

$\mathbf{Y}(z)$  is replaced by  $\mathbf{Y}(kT)$ ,

$z^{-1} \mathbf{Y}(z)$  is replaced by  $\mathbf{Y}(kT - T)$ ,

$z^{-1} \mathbf{U}(z)$  is replaced by  $\mathbf{U}(kT - T)$ ,

$\mathbf{U}(z)$  is replaced by  $\mathbf{U}(kT)$ .

$$\begin{aligned}\mathbf{Y}(kT) &= e^{-\beta T} \mathbf{Y}(kT - T) + \frac{1}{T\beta^2} [1 - e^{-\beta T} - \beta T e^{-\beta T}] \mathbf{U}(kT - T) \\ &\quad + \frac{1}{T\beta^2} [\beta T - 1 + e^{-\beta T}] \mathbf{U}(kT)\end{aligned}\quad (90)$$

Define the coefficient symbols:

$$\begin{aligned}\beta &= \frac{V_0}{L} \\ A &= \frac{1}{\beta} \left[ 1 - \frac{1}{T\beta} (1 - D) \right] \\ B &= -A + \frac{1}{\beta} (1 - D) \\ D &= e^{-\beta T}\end{aligned}\tag{91}$$

Perform the algebraic manipulation and the difference equation for the lag filter in order to derive the form found in the gust model code. [Ref. 9]

First order time lag as a function of time is:

$$\mathbf{C}_{lag}[kT] = D \cdot \mathbf{C}_{lag}[(k-1)T] + B \cdot \mathbf{E}[(k-1)T] + A \cdot \mathbf{E}[kT]\tag{92}$$

First order time lag as a function of z is:

$$\mathbf{C}_{lag}(z) = D \cdot \mathbf{C}_{lag}(z^{-1}) + B \cdot \mathbf{E}(z^{-1}) + A \cdot \mathbf{E}(z)\tag{93}$$

### Lag Lead Filter Case

The conversion of the Laplace lag lead filters to the Z-transform representation follows the lag filter case of the previous section. As before, write the lag lead block in terms of  $s$ . Then insert the sample and hold. The triangular hold transfer function and the lag lead transfer function are converted to the Z-transform representation together. Start with the combined triangular hold and lag lead transfer functions. This analysis applies to the velocity components  $v$  and  $w$  that are not in the longitudinal motion direction. The subscripts can be left off for clarity.

$$\frac{\mathbf{C}_{lag\_lead}(s)}{\mathbf{E}_{lag\_lead}(s)} = \frac{e^{Ts}(1 - e^{-Ts})^2}{Ts^2} \frac{s + \alpha}{s + \beta} \quad (94)$$

Let  $\mathbf{G}(s)$  represent  $\frac{e^{Ts}(1 - e^{-Ts})}{Ts^2} \frac{s + \alpha}{s + \beta}$ . The product of the remaining  $(1 - e^{-Ts})$  factor with the rest becomes

$$\mathbf{X}(s) = (1 - e^{-Ts})\mathbf{G}(s) \quad (95)$$

To summarize this process, if  $\mathbf{X}(s)$  involves a factor  $(1 - e^{-Ts})$ , then to obtain the Z-transform of  $\mathbf{X}(s)$ , the term  $(1 - e^{-Ts})$  may be factored out so that  $\mathbf{X}(z)$  becomes the product of  $(1 - z^{-1})$  and the Z-transform of the remaining term. As a check, remember that from the definition of  $z$  from sampled control theory,  $z = e^{sT}$ , [Ref. 15, p 150]. Make this substitution to verify the correspondence between  $(1 - e^{-Ts})$  and  $(1 - z^{-1})$ .

$$\mathbf{X}(z) = \mathbf{Z}[\mathbf{X}(s)] = (1 - z^{-1})\mathbf{Z}[\mathbf{G}(s)] \quad (96)$$

Substitute back in for  $\mathbf{G}(s)$  and repeat the process to get the Z-transform of the second  $(1 - e^{-Ts})$  factor. Take the Z-transform of the second  $(1 - e^{-Ts})$  factor and the rest:

$$(1 - z^{-1})\mathbf{Z}\left[e^{Ts}(1 - e^{-Ts})\frac{s + \alpha}{Ts^2(s + \beta)}\right]$$

In like manner to the process that pulled out the  $(1 - z^{-1})$  factor, the  $(1 - e^{-Ts})$  factor is pulled out of the Z-transform operator.

$$(1 - z^{-1})(1 - z^{-1})\mathbf{Z}\left[e^{Ts}\frac{s + \alpha}{Ts^2(s + \beta)}\right] \quad (97)$$

$e^{Ts}$  can be pulled out and converted to  $z$ . See the two pages preceding the equation (85) result in the section on the lag filter case for the details.

$$\begin{aligned}
& z(1-z^{-1})^2 \frac{1}{T} \mathbf{Z} \left[ \frac{s+\alpha}{s^2(s+\beta)} \right] \\
& z(1-z^{-1})^2 \frac{1}{T} \mathbf{Z} \left[ \frac{1}{s(s+\beta)} + \frac{\alpha}{s^2(s+\beta)} \right]
\end{aligned} \tag{98}$$

In the table of z transforms from [Ref. 15, p 50], there is an entry of the form

$$\frac{a}{s(s+a)}, \tag{99}$$

and also an entry of the form

$$\frac{a^2}{s^2(s+a)}. \tag{100}$$

Manipulate constants to get the table factor form

$$z(1-z^{-1})^2 \frac{1}{T} \mathbf{Z} \left[ \frac{1}{\beta} \frac{\beta}{s(s+\beta)} + \frac{\alpha}{\beta^2} \frac{\beta^2}{s^2(s+\beta)} \right], \tag{101}$$

and plug in the Z-transform from the table to get

$$z(1-z^{-1})^2 \frac{1}{T} \left[ \frac{1}{\beta} \frac{(1-e^{-\beta T})z^{-1}}{(1-z^{-1})(1-e^{-\beta T}z^{-1})} + \frac{\alpha}{\beta^2} \frac{[(\beta T - 1 + e^{-\beta T}) + (1-e^{-\beta T} - \beta T e^{-\beta T})z^{-1}]z^{-1}}{(1-z^{-1})^2(1-e^{-\beta T}z^{-1})} \right].$$

Simplify and rearrange.

$$\begin{aligned}
& (1-z^{-1}) \frac{1}{T} \left[ \frac{1}{\beta} \frac{(1-e^{-\beta T})}{(1-e^{-\beta T}z^{-1})} + \frac{\alpha}{\beta^2} \frac{[(\beta T - 1 + e^{-\beta T}) + (1-e^{-\beta T} - \beta T e^{-\beta T})z^{-1}]}{(1-z^{-1})(1-e^{-\beta T}z^{-1})} \right] \\
& \frac{1}{\beta^2 T} \left[ \frac{\beta(1-z^{-1})(1-e^{-\beta T}) + \alpha[(\beta T - 1 + e^{-\beta T}) + (1-e^{-\beta T} - \beta T e^{-\beta T})z^{-1}]}{1-e^{-\beta T}z^{-1}} \right].
\end{aligned} \tag{102}$$

The following is taken from the definition of transfer function:

$$\frac{Output(z)}{Input(z)} = \text{the transfer function}$$

$$\frac{\mathbf{Y}(z)}{\mathbf{U}(z)} = \frac{1}{T\beta^2} \left[ \frac{\beta(1-z^{-1})(1-e^{-\beta T}) + \alpha[(\beta T - 1 + e^{-\beta T}) + (1-e^{-\beta T} - \beta T e^{-\beta T})z^{-1}]}{1-e^{-\beta T}z^{-1}} \right].$$

Arrange terms using algebra.

$$\begin{aligned} \mathbf{Y}(z)(1-e^{-\beta T}z^{-1}) &= \frac{1}{T\beta^2} [\alpha(1-e^{-\beta T} - \beta T e^{-\beta T}) + \beta(1-e^{-\beta T} - (1-e^{-\beta T}))] z^{-1} \mathbf{U}(z) \\ &\quad + \frac{1}{T\beta^2} \alpha [\beta T - (1-e^{-\beta T})] \mathbf{U}(z) \\ \mathbf{Y}(z) &= e^{-\beta T} z^{-1} \mathbf{Y}(z) + \frac{1}{T\beta^2} \alpha (1-e^{-\beta T} - \beta T e^{-\beta T}) z^{-1} \mathbf{U}(z) \\ &\quad + \frac{1}{T\beta^2} \alpha [\beta T - (1-e^{-\beta T})] \mathbf{U}(z) \end{aligned} \tag{103}$$

Using the shifting theorem of the Z-transform,

$\mathbf{Y}(z)$  is replaced by  $\mathbf{Y}(kT)$ ,

$z^{-1}\mathbf{Y}(z)$  is replaced by  $\mathbf{Y}(kT - T)$ ,

$z^{-1}\mathbf{U}(z)$  is replaced by  $\mathbf{U}(kT - T)$ ,

$\mathbf{U}(z)$  is replaced by  $\mathbf{U}(kT)$ , and

$$\begin{aligned} \mathbf{Y}(kT) &= e^{-\beta T} \mathbf{Y}[(k-1)T] + \frac{\alpha}{T\beta^2} (1-e^{-\beta T} - \beta T e^{-\beta T}) \mathbf{U}[(k-1)T] \\ &\quad + \frac{\alpha}{T\beta^2} (\beta T - 1 + e^{-\beta T}) \mathbf{U}(kT) \end{aligned} \tag{104}$$

where  $k$  is the frame number.

Define the coefficient symbols:

$$\begin{aligned}
\beta &= \frac{V_0}{L} \\
A &= \frac{1}{\beta} \left[ 1 - \frac{1}{T\beta} (1 - D) \right] \\
D &= e^{-\beta T} \\
I &= \frac{1}{\sqrt{3}} + \left( \frac{1 - \frac{1}{\sqrt{3}}}{\beta T} \right) (1 - D) \\
J &= -I + \frac{\alpha}{\beta} (1 - D)
\end{aligned} \tag{105}$$

Perform the algebraic manipulation and the difference equation for the lag filter in order to derive the form found in [Ref. 9] and the gust model code.

First order lag lead as a function of time is:

$$\mathbf{C}_{lag\_lead}[kT] = D \cdot \mathbf{C}_{lag\_lead}[(k-1)T] + J \cdot \mathbf{E}[(k-1)T] + I \cdot \mathbf{E}[kT] \tag{106}$$

First order lag lead as a function of z is:

$$\mathbf{C}_{lag\_lead}(z) = D \cdot \mathbf{C}_{lag\_lead}(z^{-1}) + J \cdot \mathbf{E}(z^{-1}) + I \cdot \mathbf{E}(z) \tag{107}$$

## *Random Number Generator*

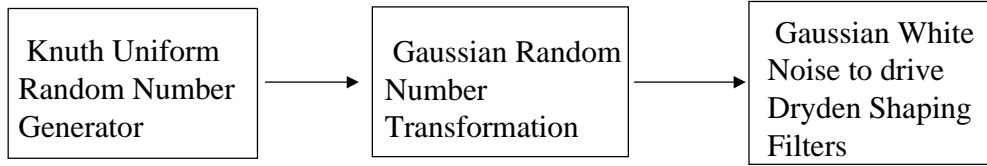


Figure 48. Uniform random numbers modified to produce Gaussian white noise.

A portable uniform random number generator is selected from [Ref. 17, p 283] originally written by D. Knuth. This replaces the random number generator used in the Airlab simulation, RANFP, which was found to have problems. Airlab did not have a great need for gust response and the random number generator was not given much attention when the software was converted from the CDC NOS machine to the VAX 750. Periodicity in some observed control parameters pointed to a problem with the Dryden gust model. The periodicity was traced to the random number generator RANFP. A plot of a 400 number sequence of random numbers produced by RANFP followed by 400 numbers from the D. Knuth generator “ran3Knuth” is shown for comparison.



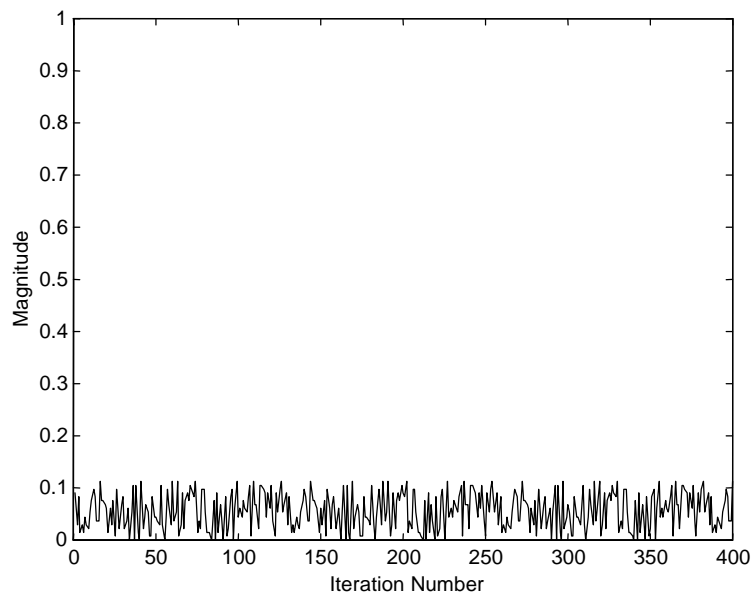


Figure 49. Uniform random numbers from RANFP, 400 values.

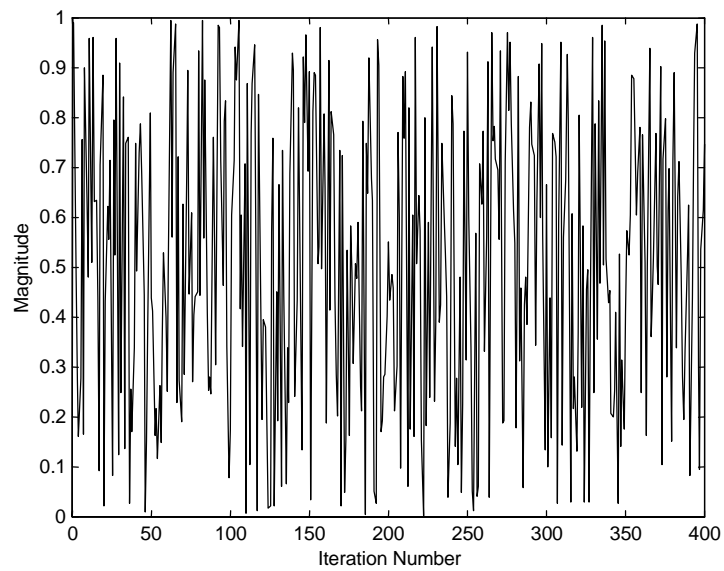


Figure 50. Uniform random numbers from ran3Knuth, 400 values.

Except for an unexplained factor of 10 scale decrease, the RANFP doesn't look too bad at first glance. However, if the first 200 numbers are paired with the second 200 numbers and plotted as an  $(x,y)$  coordinate pair, the following non-random behavior is produced by RANFP. Notice that figure 51 and 52 do not have the same scale.

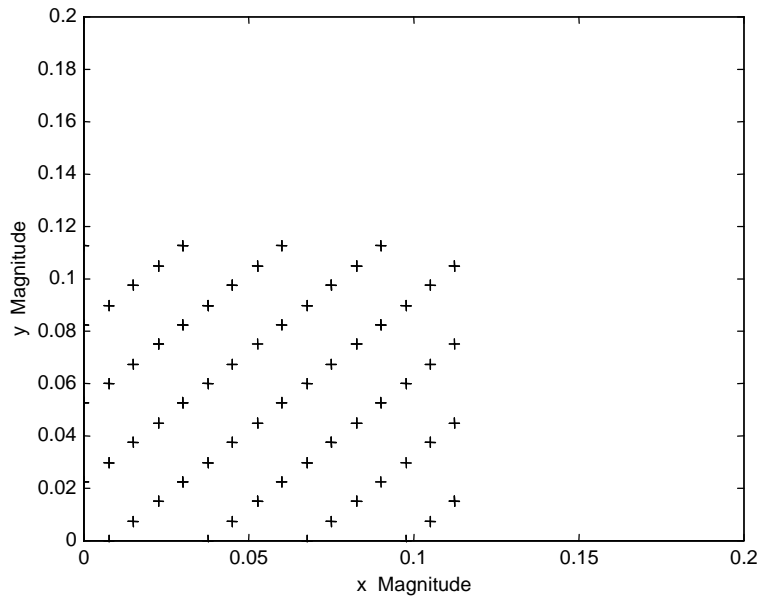


Figure 51. Pairs of uniform random numbers from RANFP.

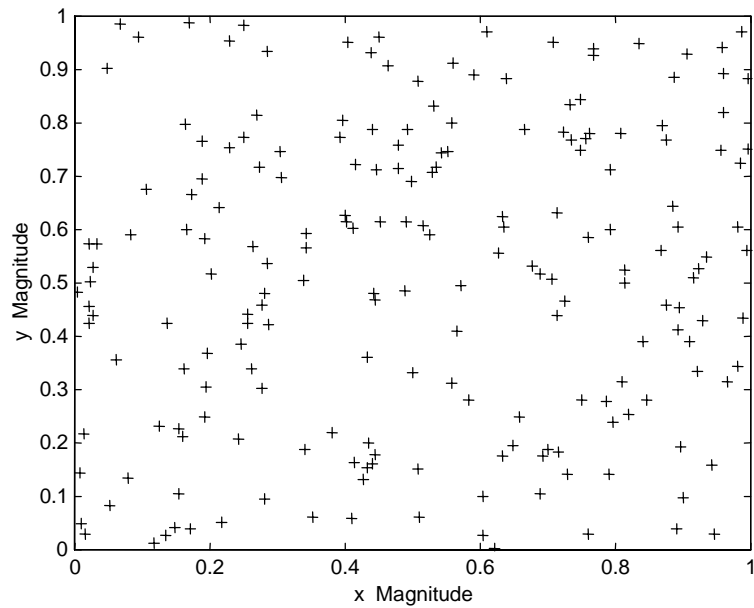


Figure 52. Pairs of uniform random numbers from ran3Knuth.

The Knuth generator, ran3Knuth, based on ran3 from [Ref. 17, p 282], is designed to be portable across different computer systems. This implementation has been verified to work with Sun FORTRAN, and Sun as well as Windows95 PC C, and Simulink, and C code generated from Simulink Real-Time Workshop.

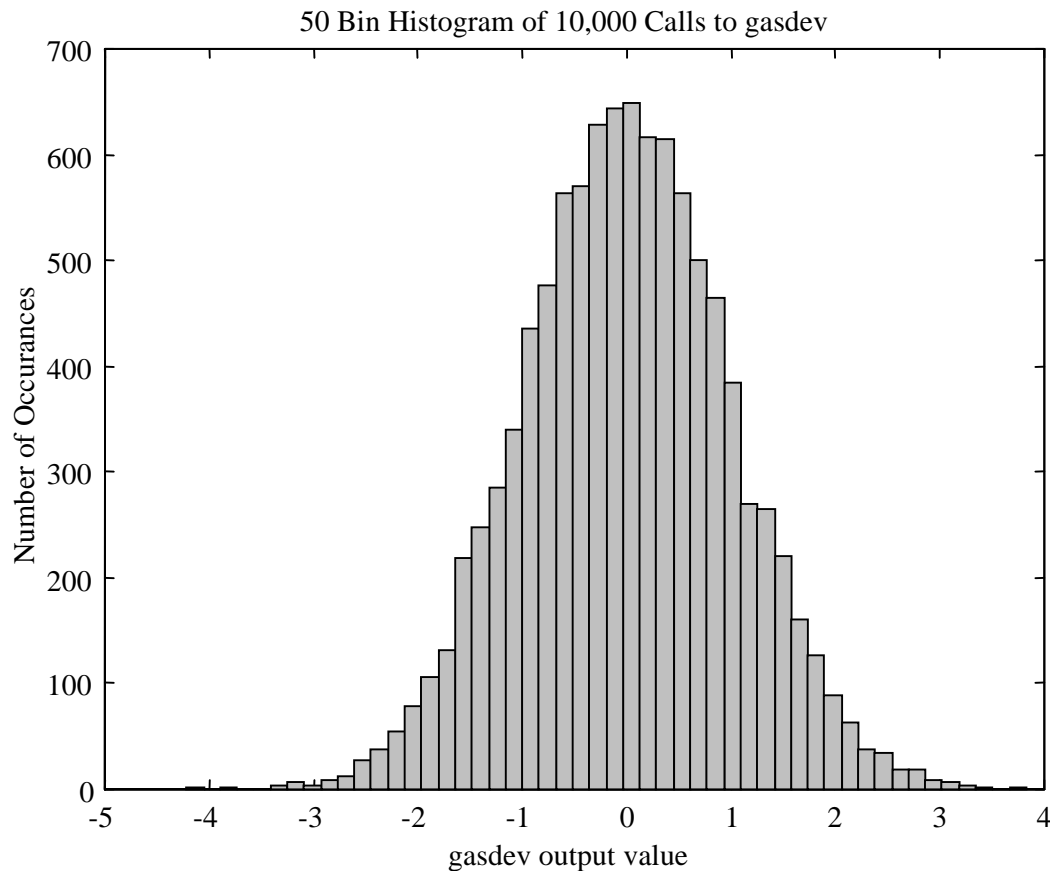


Figure 53. A 50 bin histogram of 10000 Gaussian distributed random numbers from gasdev.

The uniform random numbers need to be modified to generate the required Gaussian distributed white noise for the Dryden gust model. The Box-Muller transformation method is used to generate a Gaussian random number from a uniform random number. Subroutine “gasdev” from [Ref. 17, p 289] is used to generate Gaussian random numbers in the B-737 Linear Autoland Simulink Model. Figure 53 shows the expected bell-shaped curve Gaussian character of the numbers generated. Figure 54 is a plot of pairs of the Gaussian random numbers and is evidence of the lack of unwanted periodicity.

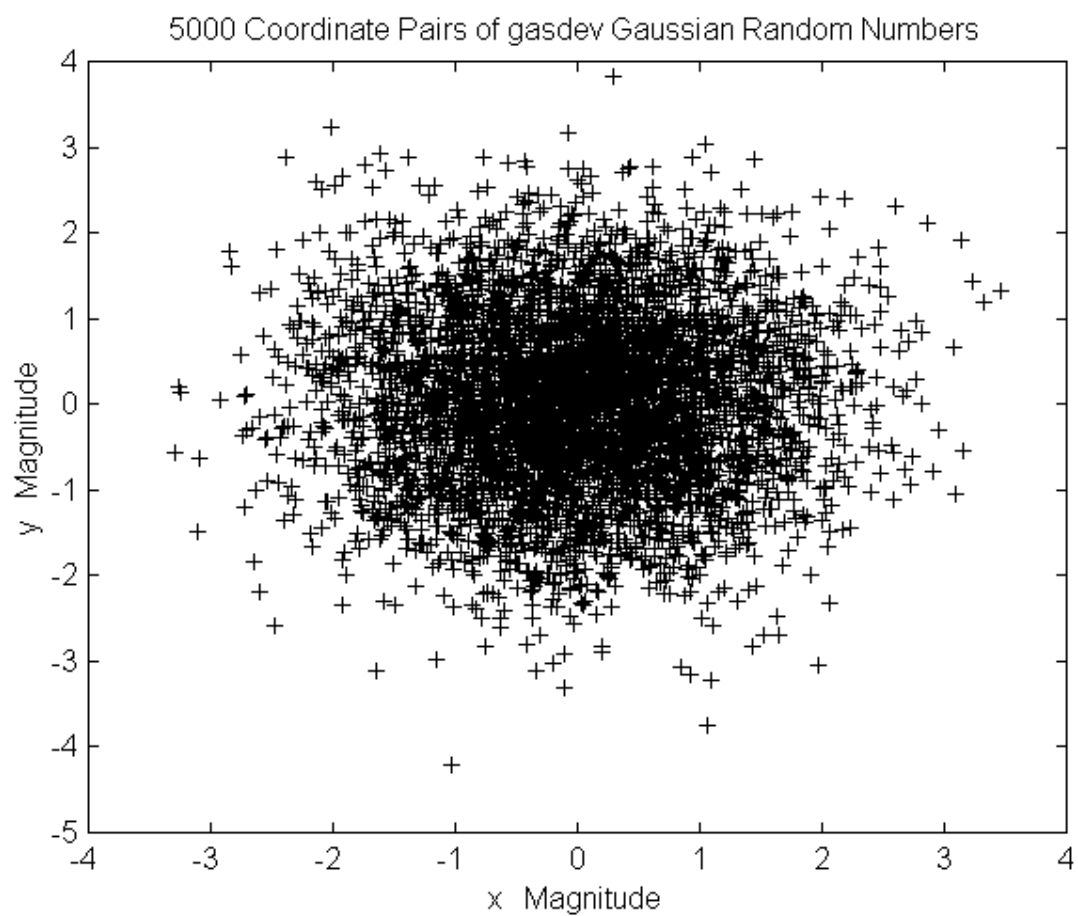


Figure 54. 5000 pairs of Gaussian random numbers from gasdev.

# Autopilot

## Control Concepts

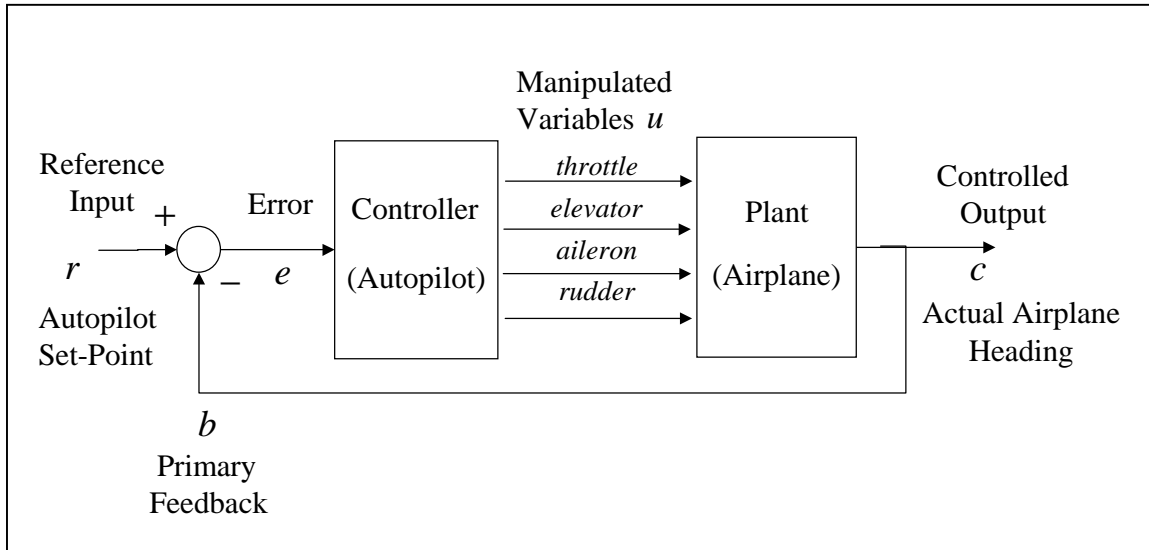


Figure 55. Basic control terminology adapted from chapter 2 [DiStefano III].

- Control Set Point

A controller and a plant use feedback to maintain a desired state of the system. The desired state of the system is fed in through an input and compared with the present state of the system in a summing junction. The difference in the two values is the error signal. This error signal is sent to the controller which calculates the “manipulated variables” so as to move the plant closer to the desired set point. The state of the system is sent back to be compared with the input and is output as the new state of the system. This completes the “feedback” loop.

- Control Error

“Error” used in the feedback control context means the difference between the commanded and preexisting state of the system. It is not used in the sense of being a mistake or system defect. The controller attempts to minimize this error signal.

- Localizer Error

The lateral autopilot performs its function by minimizing the track angle error and localizer error using the ailerons and spoilers. The angle from true north to the runway centerline is subtracted from the angle from true north to the aircraft position track and this becomes the track angle error. This difference is the aircraft track angle misalignment with the runway. This is the error that prompts most of the lateral control action to change the heading angle to align the aircraft track with the runway. Once the aircraft is close enough to the localizer runway radial (also the runway centerline), a finer heading correction that uses the localizer ILS radio signal deviation begins. This signal partially opposes the track angle error to prevent the aileron control action from causing the aircraft to overshoot the runway centerline. The track angle error effect on the heading still predominates that continues the parallel alignment with the runway. Fine corrections by the localizer radio signal are added to the track angle error to keep the aircraft on the center of the runway until landing.

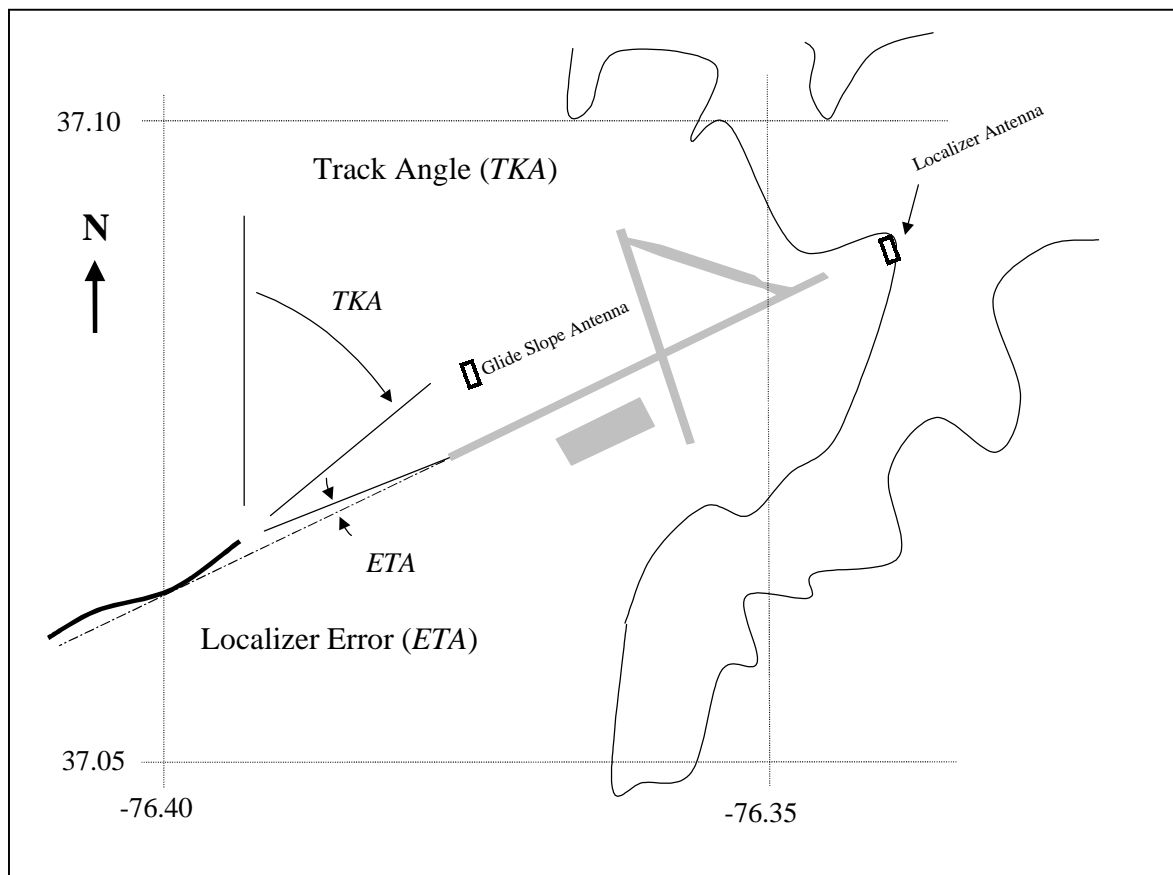


Figure 56. Track angle and error in track angle geometry relative to the airport runway.

- **Glide Slope Error**

The longitudinal autopilot attempts to minimize the difference between the flight path angle to the glide slope ILS transmitter and the desired glide slope angle. This glide slope error is the set point for the pitch control. The elevator and stabilator are used to minimize this error. This is prevented once the flare altitude is reached and a preprogrammed sink rate and pitch is commanded using altitude as a reference.

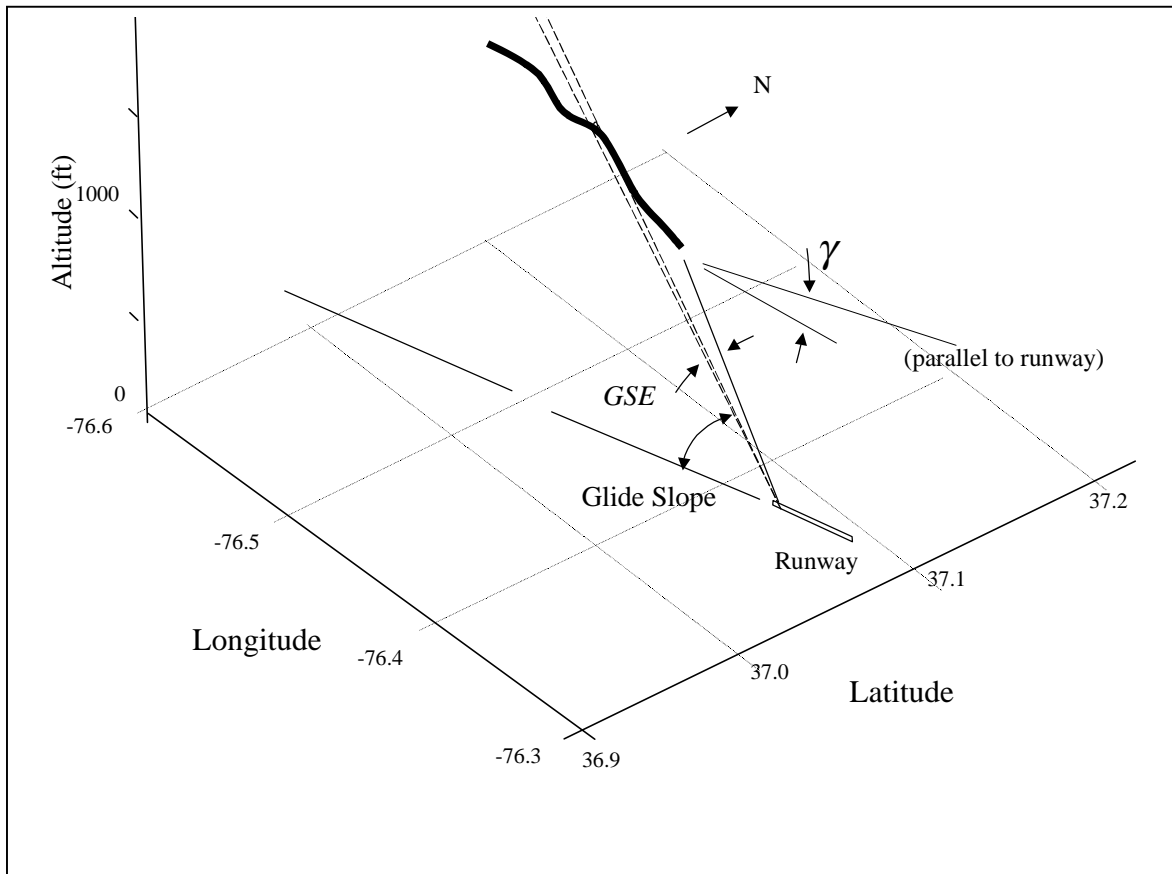


Figure 57. Glide slope error ( *GSE* ) and flight path angle  $\gamma$ .

- **Manipulated Variables**

The variables that a controller uses to change the state of the plant are the “manipulated variables”. A very simple airplane is completely controlled using the throttle, aileron, elevator, and rudder. The B-737 has additional control surfaces and automatic systems, though the throttle, aileron, elevator, and rudder are the most important.

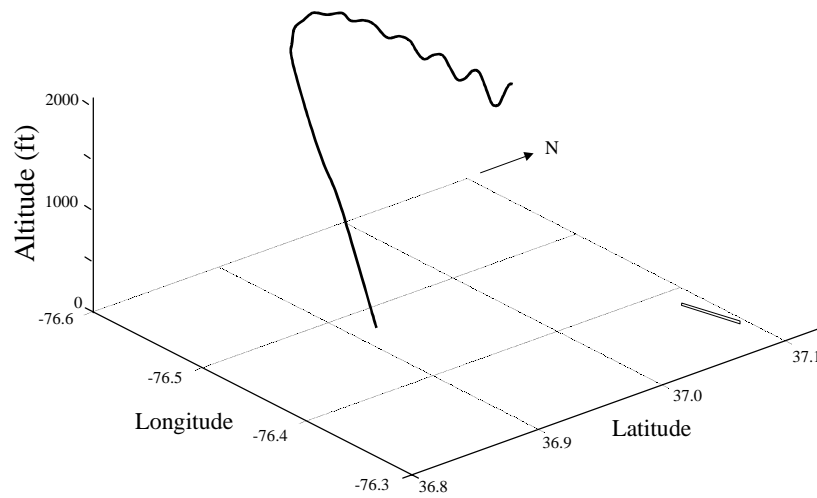


Figure 58. Flight path with fixed command inputs.

- Natural Frequency

A physical system described by differential equations such as an aircraft will have a characteristic frequency that arises from a displacement from equilibrium. In the case of the aircraft airframe model, the equilibrium for the control system is straight and level flight. If the aircraft is perturbed from level flight, and the design is a stable one, it will oscillate about the straight and level path as shown. This is true for the case with no restoring control inputs called “open loop”. If there are restoring control inputs attempting to maintain a course as an autopilot does, then the “natural frequency” of the “natural response” is hidden by the action of the control system.

- Stability

A physical system is considered “stable” if a bounded input produces a bounded output. Another type of stability is that if a system is displaced from equilibrium, it will return to equilibrium after oscillations at the “natural frequency” which decrease in amplitude. A system is “unstable” if a displacement from equilibrium causes further movement away from equilibrium. If unstable, the oscillations increase in amplitude. For an aircraft airframe, the tendency to return to a straight course is related to the relative position of the center of pressure and the center of mass (gravity). The center of mass needs to be in front of the center of pressure to naturally keep the front of the vehicle facing the apparent wind as demonstrated by the action of a wind vane. Control surfaces must counteract this tendency toward straight flight to cause the aircraft to turn. Sometimes other design considerations trade off wind vane stability in favor of maneuverability and stability augmentation is used. The yaw damper is a lateral control system stability augmentation system.



- Symmetry

The symmetry of the typical airplane allows separation of the control of the longitudinal motion from the control of the lateral motion [Ref. 20, p 48]. The symmetry reduces every rolling torque and side force to zero. This approximation still is good for small turns and sideslips. The B-737 autopilot makes this approximation and separates the longitudinal control system from the lateral control system.

- Rate Control

The aileron is a rate control device. The angular velocity of the roll is proportional to the angular deflection of the aileron control wheel. As long as the control wheel is deflected, the airplane will roll with a constant angular rate.

- Position Control

The elevator is a position control device. The angle of pitch is proportional to the yoke deflection. When the yoke is deflected, the airplane will orient to the new pitch orientation and remain there. The pitch angle will not keep changing for a constant yoke deflection.

### Autothrottle

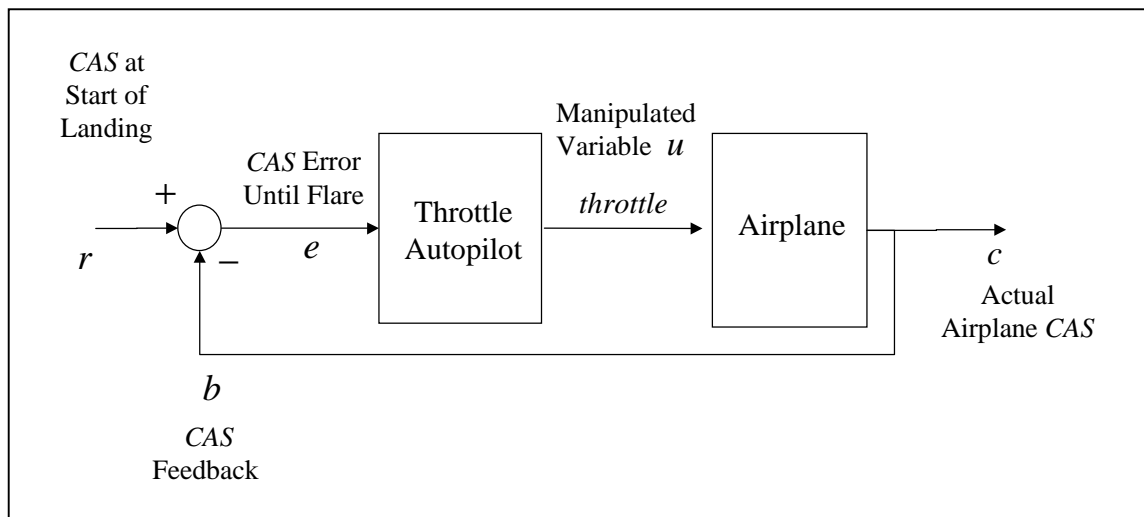


Figure 59. Autothrottle feedback loop.

The autothrottle keeps the airspeed the same until flare, then it decreases. The elevator is used to keep the airplane on the glide slope, and the autothrottle does what is necessary to keep the calibrated airspeed (CAS) the same coming down the glide slope.



## Lateral Autopilot

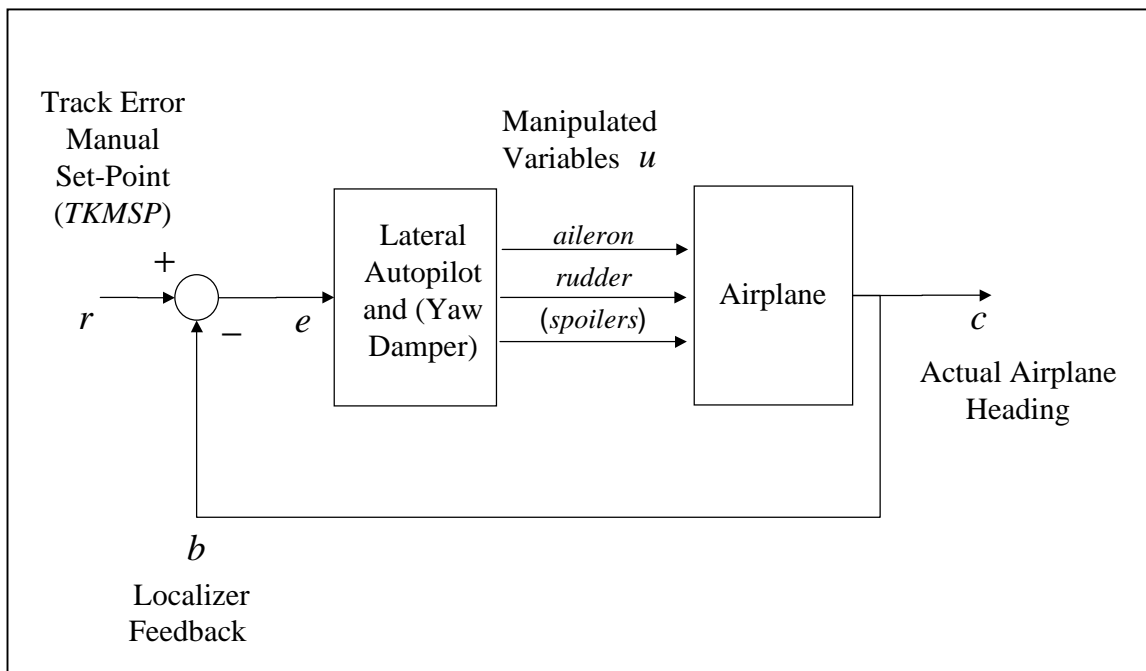


Figure 61. Lateral autopilot feedback loop.

The lateral autopilot primarily uses the ailerons and spoilers to roll the aircraft. The spoilers are used if a steep bank or one of long duration is commanded. They counteract the tendency of the aircraft to have roll adverse yaw. The lateral autopilot controls yaw angle as well as the roll angle due to tight coupling of the rolling motion to the yawing motion. If additional correction is necessary to control sideslip, the rudder is manipulated. The rudder is active during gusts through action of the yaw damper, but there is usually no rudder command associated with the rudder movement in yaw damping. The action of the roll control is rate control. A constant angle of aileron deflection results in a constant rate of roll. A bank requires a positive deflection held for a short time, followed by a negative deflection to stop the rolling rate once the angle of the desired bank is achieved. Ailerons produce a drag differential with the down aileron having more drag than the up flap. This tends to yaw the nose of the aircraft in the direction opposed to the desired bank. The spoiler of the low wing deploys to reduce the lift and increase the drag to counteract this tendency for aileron adverse yaw [Ref. 6, p 89-90].

### *Lateral Autopilot Simulink Diagram*

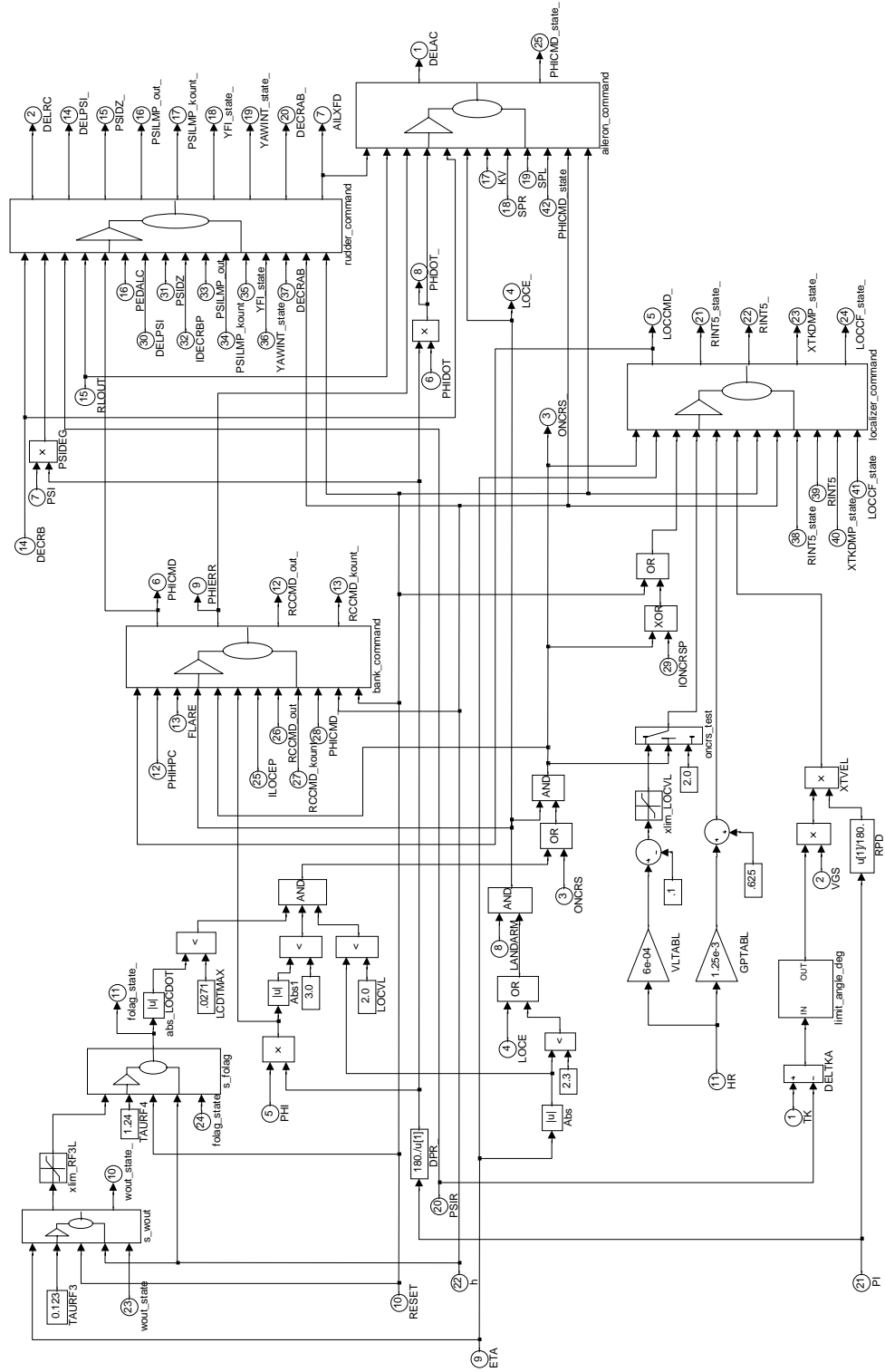


Figure 62. Lateral autopilot Simulink diagram.

## Longitudinal Autopilot

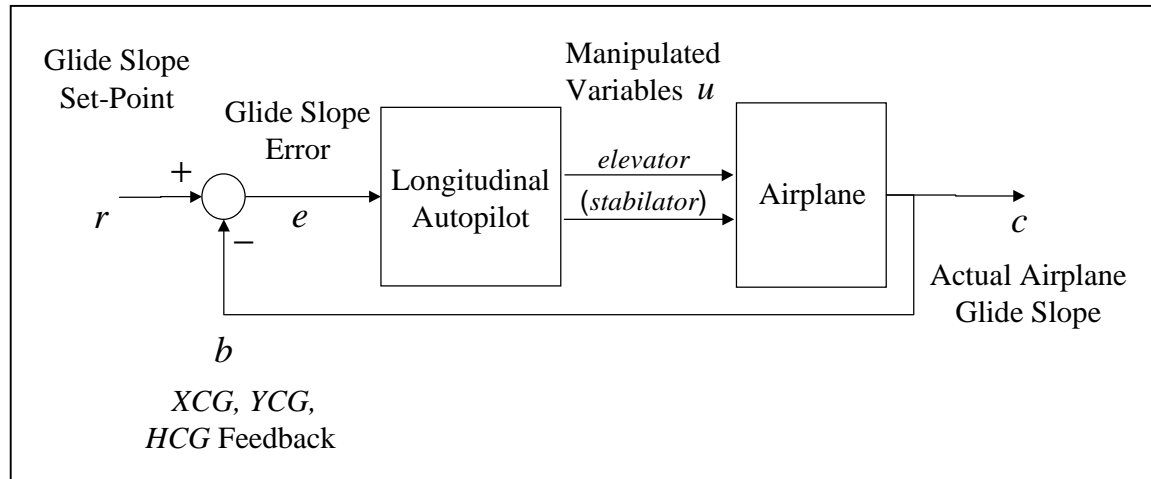


Figure 63. Longitudinal autopilot feedback loop.

The longitudinal autopilot controls the pitch orientation of the aircraft. The elevator is the primary pitch control device. A slower acting automatic trim system tries to drive the elevator back to zero by moving the horizontal tail surface to which the elevator is attached. This moving assembly is called the stabilator. The action of the pitch control is position control. The aircraft pitches until equilibrium is reached for a given deflection of the elevator. This acts differently than the aileron roll control. The longitudinal autopilot has two major control modes. Vertical path control (VPC) is used to hold a given altitude during normal flight. Once glide slope is engaged and a final Instrument Landing System (ILS) approach begins, the longitudinal autopilot switches to Autoland mode. The glide slope error is used to close the pitch control loop. The autopilot commands pitch down if the aircraft is coming in too high; pitch up if too low. This continues until the altitude where the flare mode begins is attained. Then, a preprogrammed pitch profile for the flare is commanded.

*Longitudinal Autopilot Simulink Diagram*

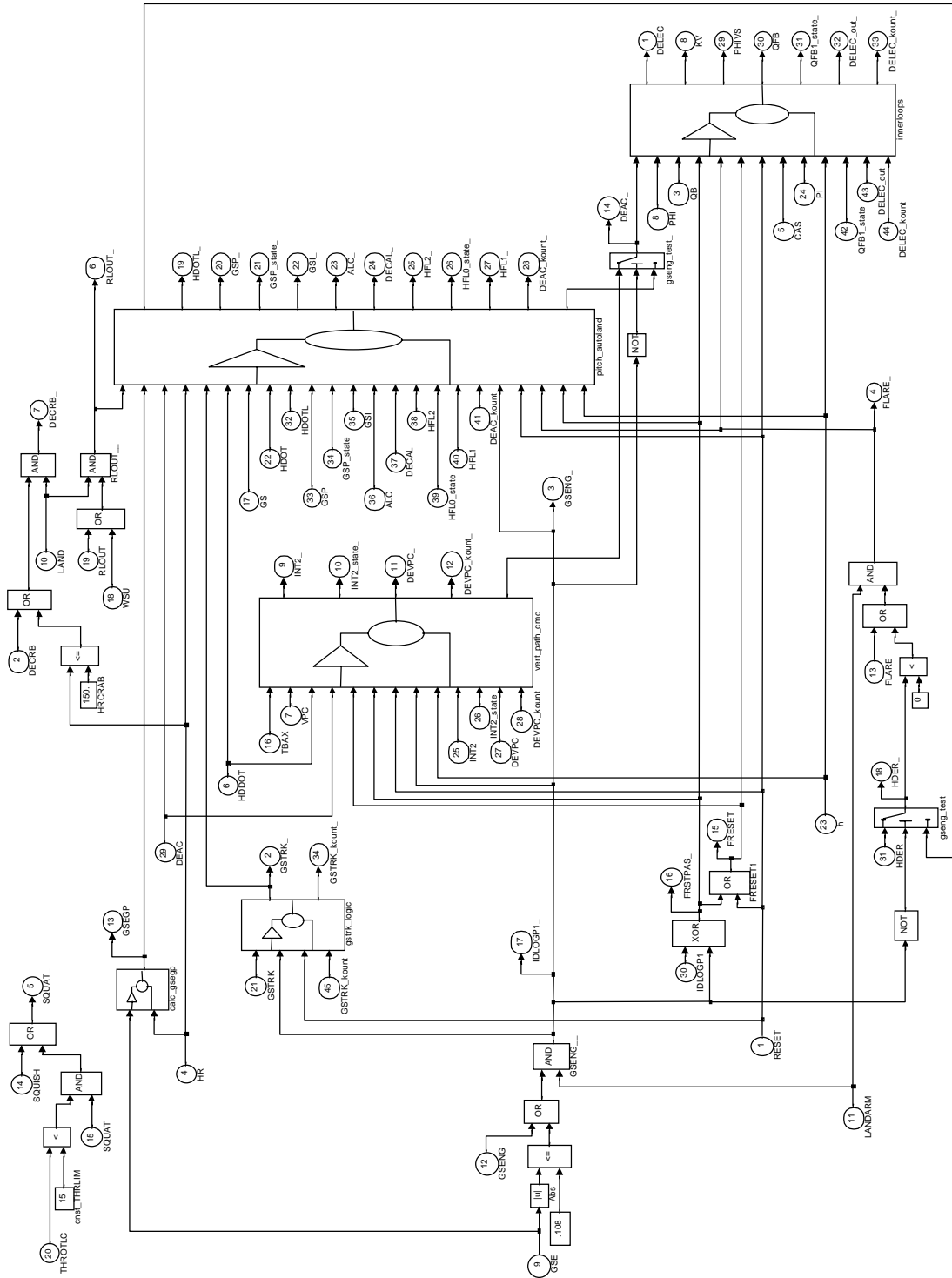


Figure 64. Longitudinal autopilot Simulink diagram.

### Vertical Path Control

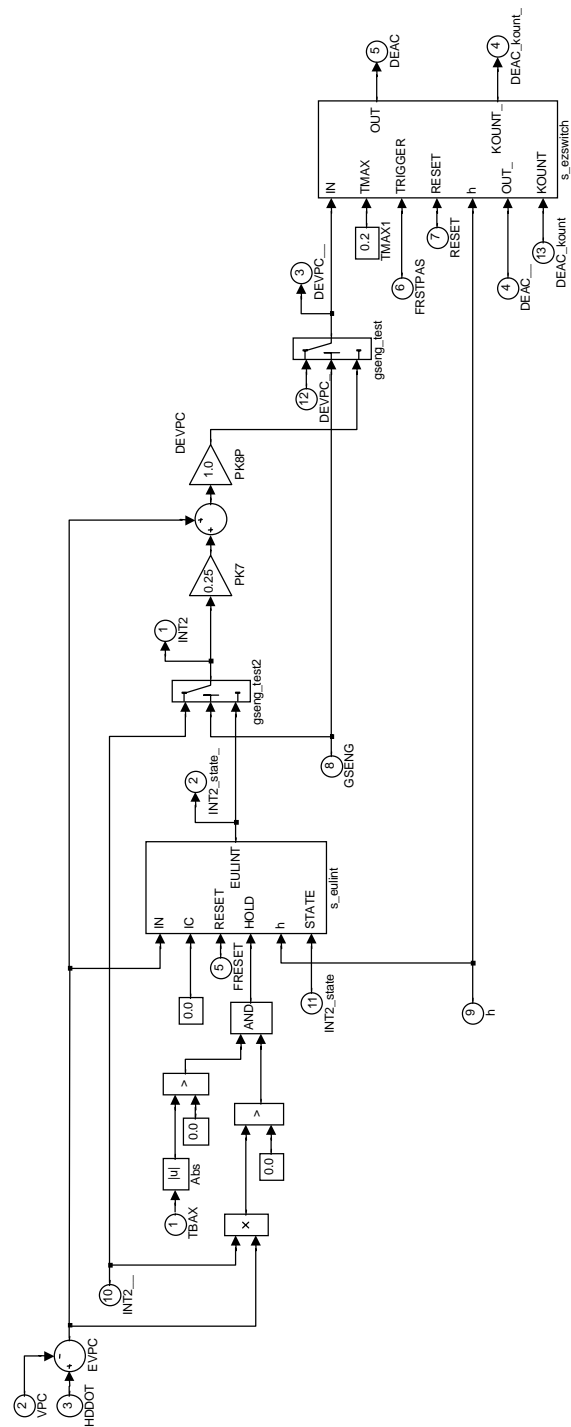


Figure 65. Longitudinal autopilot vertical path control law Simulink diagram.

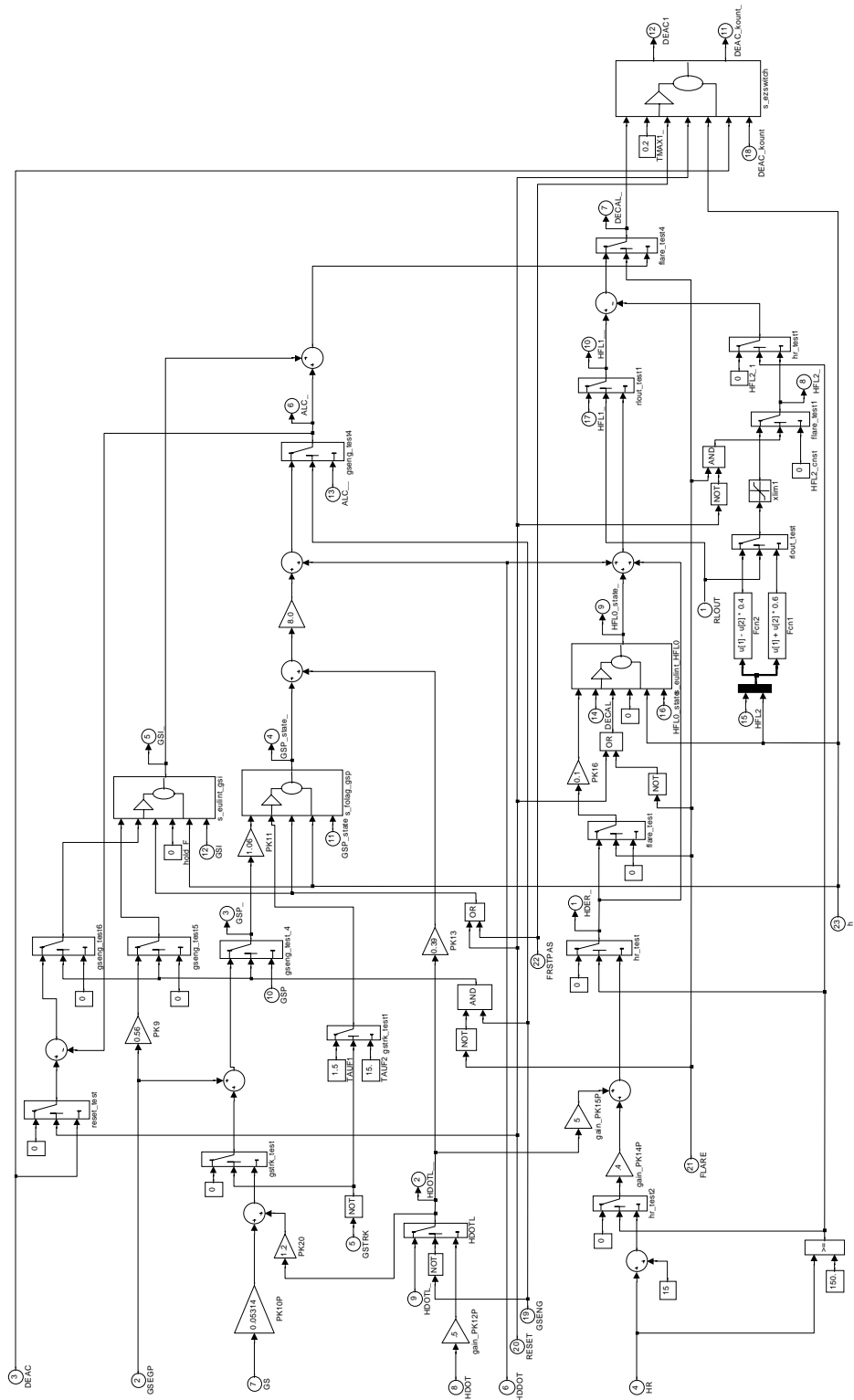


Figure 66. Longitudinal autopilot Autoland control law Simulink diagram.



## Flight Modes

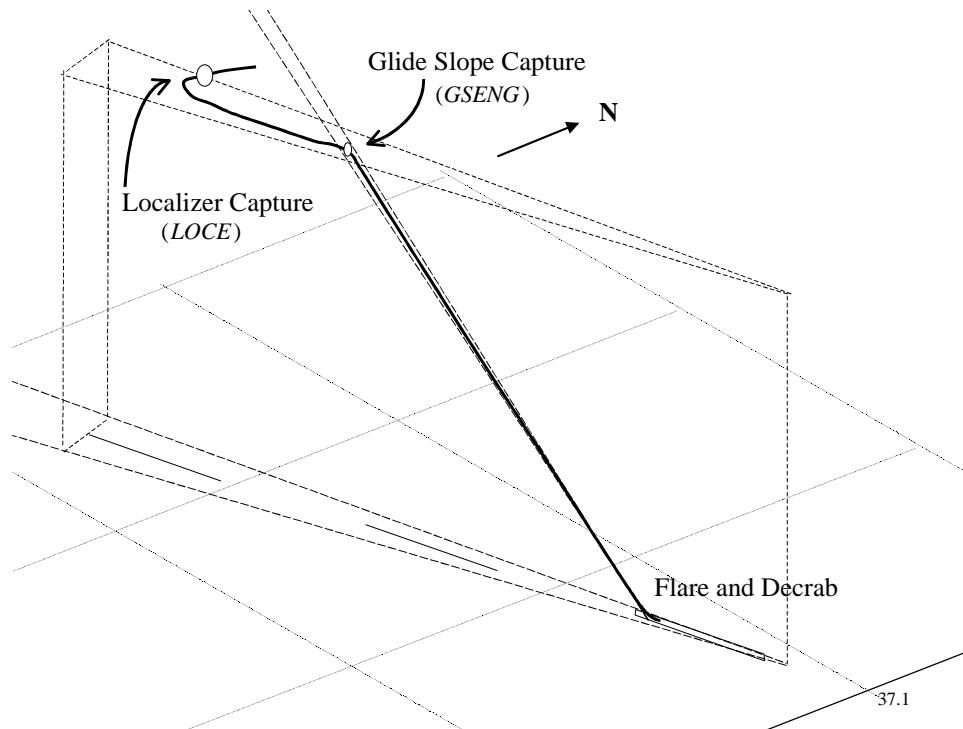


Figure 67. Autopilot mode changes along the approach path.

- Localizer Engaged

The localizer engaged mode is entered when the track error is less than 2.3 degrees and the aircraft is within 6500 feet of the runway threshold along the runway Y-axis either side. The LANDARM flag indicates the Y-axis condition.

- Glide Slope Engaged

The glide slope engaged mode is entered when the glide slope error is less than .108 degrees from the ILS glide slope and the aircraft is within 6500 feet of the runway threshold. LANDARM is true as for the localizer engaged mode.

- Decrab

The roll and yaw axes are controlled to decrease the “crab” angle to the runway in a crosswind landing. The decrab mode is also called the runway alignment mode. The landing gear are made to line up better with the runway by the aircraft executing a sideslip into the wind. Decrab mode begins when the aircraft has descended so that the landing gear are less than 150 feet above the runway.

- Flare

The FLARE flag is set when the airplane descends below 50 feet above the runway and LANDARM is true. A preprogrammed pitch up rate profile is commanded until the wheels are on the ground. The pitch up rate depends upon the aircraft sink rate. When the wheels are on the ground and spun up, rollout begins and the aircraft pitches down. The simulation stops before rollout for the Airlabs version of B-737 Linear Autoland.

- Rollout

Rollout is a mode that begins when the wheels are on the ground and have spun up. The Airlabs version of the B-737 Linear Autoland stops before this occurs.

## Engine Model

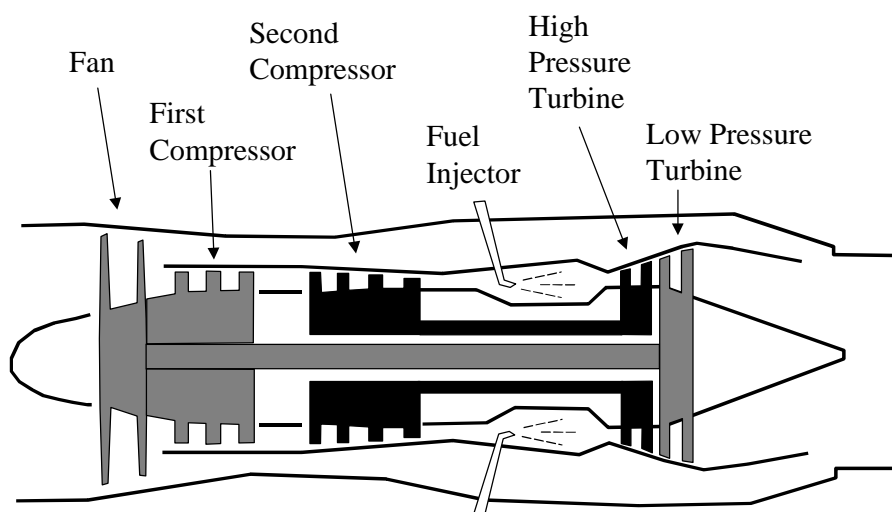


Figure 68. Turbofan engine.

The engine used in the B-737 linear Autoland model is the Pratt and Whitney JT8D-7. It is an example of the turbofan type of jet engine. A jet engine works by compressing the air and injecting fuel into a combustion chamber. The increase in temperature from the burning of the fuel increases the energy content of the air. This hot air is expanded through a turbine to provide energy to compress new intake air. The energy remaining in the hot air is converted to kinetic energy of motion of the air expelled from the nozzle. A turbofan engine has two compressors in front of the combustion chamber and two turbines following it. Some air from the first compressor is ducted around the combustion chamber and turbines and is called “bypass air”. This is done to increase fuel efficiency by reducing the speed of the nozzle flow but increasing the mass flow.

The primary control parameter of the jet engine is the fuel flow rate into the combustion chamber. The throttle setting controls this fuel flow. The throttle setting is transferred using a cable arrangement to a fuel control unit power lever. The setting of the power lever is called the “cross shaft angle” (CSA).

CSA, Cross shaft angle is the “power lever” fuel flow setting on the fuel control unit for the engine.

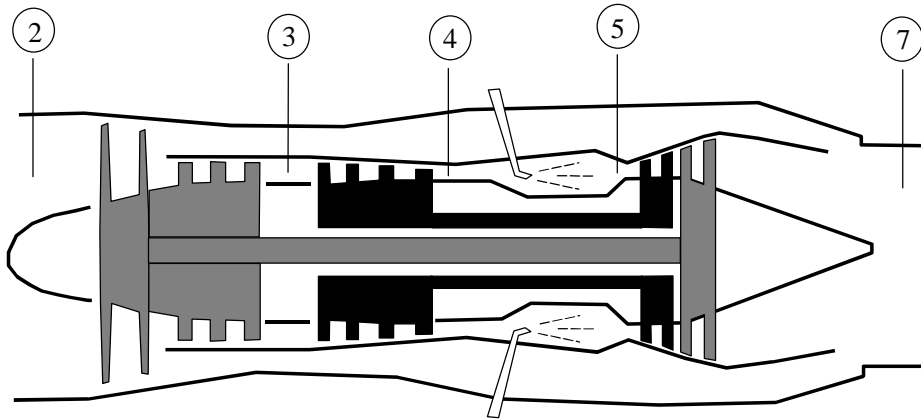


Figure 69. Engine parameter stations.

The equations describing the operation of a turbofan engine use a standard nomenclature of stations from the entrance of the engine to the exit [Ref. 19, p 153]. Station 2 is at the inlet to the engine. Station 3 is at the exit of the first compressor. Station 4 is at the exit of the second turbine. Station 5 is at the entrance to the first turbine. Station 7 is at the exit of the primary nozzle. In [Ref. 13, chapter 1 and 2] is a discussion of the thermodynamic equations that pertain to the turbofan and other jet engines. The station numbering convention is slightly different. Also, many of the equations are expressed in terms of pressure and temperature ratios. The B-737 linear Autoland engine model uses pressure and temperature ratios too, however a different reference is used. [Ref. 13, p 2] uses flight level ambient as the reference  $p$  whereas the B-737 linear Autoland uses the International Standard Atmosphere (ISA) value for some of its parameters. The temperature reference is called sea level temperature *SLTEMP*. Many aircraft parameters for instrumentation use this reference convention to standardize table descriptions [Ref. 2, p 4-3]. The ideal gas law ratio of specific heats for air is used in the equations.

The nomenclature for the thermodynamics of the engine overlaps aeronautics usage for several variables.

Table 9. Engine Symbols that may Cause Confusion with Control Parameter with the Same Name.

Symbol	Engine Context	Previous Definition
$\gamma$	ratio of specific heats of air	flight path angle
$\delta$	ratio of engine station pressure with a reference	control surface deflection with a subscript denoting which surface
$\theta$	ratio of engine station temperature with a reference	Euler pitch angle

### ***International Standard Atmosphere (ISA)***

29.92 in. Hg, ISA Sea Level Pressure, 15 deg. C (288.16 deg. Kelvin), ISA Sea Level Temperature [Ref. 2, p 4-3]

Table 10. Engine Nomenclature used in the B-737 Linear Autoland Model

Symbol	Symbol in Model	Definition
	<i>SLTEMP</i>	ISA Sea level temperature
$\theta_{ambient} SLTEMP$	<i>SAT</i>	Static air temperature at flight level ambient
$\theta_{ambient}$	<i>THETAMB</i>	$\frac{\text{stagnation temperature flight level ambient}}{SLTEMP}$
$P_{t0}$		Total air pressure at flight level ambient conditions. It is also called “stagnation pressure”, the pressure reached if the flowing stream is stopped without any heat transfer to the surroundings (adiabatic).
$P_0$		Static air pressure at flight level ambient conditions. Pressure measured in a tube perpendicular to the free stream.
$\delta_{ambient}$	<i>DELTAMB</i>	$\frac{P_{t0}}{\text{ISA sea level static pressure}}$
$c_p$		Specific heat at constant pressure $\frac{\text{heat energy supplied}}{(\text{temperature rise}) (\text{mass of air held at constant pressure})}$
$c_v$		Specific heat at constant volume $\frac{\text{heat energy supplied}}{(\text{temperature rise}) (\text{mass of air held at constant volume})}$
$\gamma$		Ratio of specific heats in air. $\gamma = \frac{c_p}{c_v} = 1.40$

## Engine Pressure Ratio

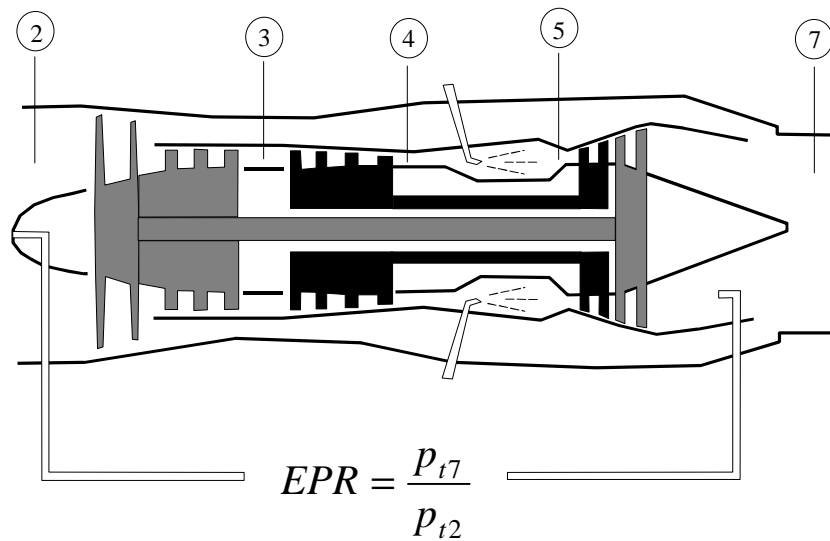


Figure 70. Engine pressure ratio.

Engine pressure ratio is the ratio of the compressor inlet pressure to the nozzle exit pressure. It is the primary feedback parameter that is used to set thrust. It is measured using small pitot tubes in front of the compressor and behind the nozzle. The pressure altitude and ambient air temperature affect EPR settings. The amount of compressor bleed air being used to prevent compressor stall, anti-icing and turbine cooling will also affect the setting.

Table 11. Additional Engine Nomenclature used in the B-737 Linear Autoland Model

Symbol	Symbol in Model	Defintion
	<i>EPRC</i>	Commanded engine pressure ratio, a function of cross-shaft angle in degrees, bleed valve status, altitude in feet, Mach number, and engine inlet temperature in degrees Fahrenheit <i>TT2F</i> . Calculation of <i>EPRC</i> is from a polynomial in powers of <i>CSA</i> and <i>TT2F</i>
	<i>EREF</i>	The <i>EPR</i> present at the start of a spool-up. This is a meaning specific to this B-737 model. It is a local variable intended to smooth a transition, not an <i>EPR</i> limit as in current usage. <CAUTION> The more current meaning of <i>EREF</i> is <i>EPR</i> reference limit. The pressure ratio setting that limits the power to the maximum power setting allowed for the current stage of flight. This is not the meaning in the B-737 model.
	<i>EREFS</i>	<i>EPR</i> existing at the start of a spool-up, expressed as a fraction of the <i>EPR</i> range.
	<i>EPRS</i>	The first deviation of the <i>EPR</i> from the idle <i>EPR</i> ( <i>EPRMIN</i> ), expressed as a fraction of the <i>EPR</i> range. $EPRS = \frac{EPR - EPR_{MIN}}{2.23 - EPR_{MIN}}$
	<i>DEPRC</i>	Difference between commanded <i>EPR</i> and actual <i>EPR</i> .
$T_{t2}$		Engine inlet total temperature
	<i>TTR</i>	Engine inlet total temperature ratio $TTR = \frac{T_{t2}}{SAT}$ also: $TTR = 1 + \frac{1}{2}(\gamma - 1)M_2^2$

The simplified engine model shown receives a throttle input through the CSA table lookup. The total temperature, aircraft altitude, and Mach number feed the commanded engine pressure (*EPRC*) ratio calculation. The engine dynamics modifies this based on previous engine EPR profile and bleed valve status and the corrected EPR is sent to the thrust correction for the airframe and flight ambient conditions to get net thrust for the airframe model. The thrust correction module is *THCORF* and corrects the thrust using a polynomial in powers of Mach number. The spool up detection sets the upper limit on the rate of change of the EPR in the engine dynamics module.

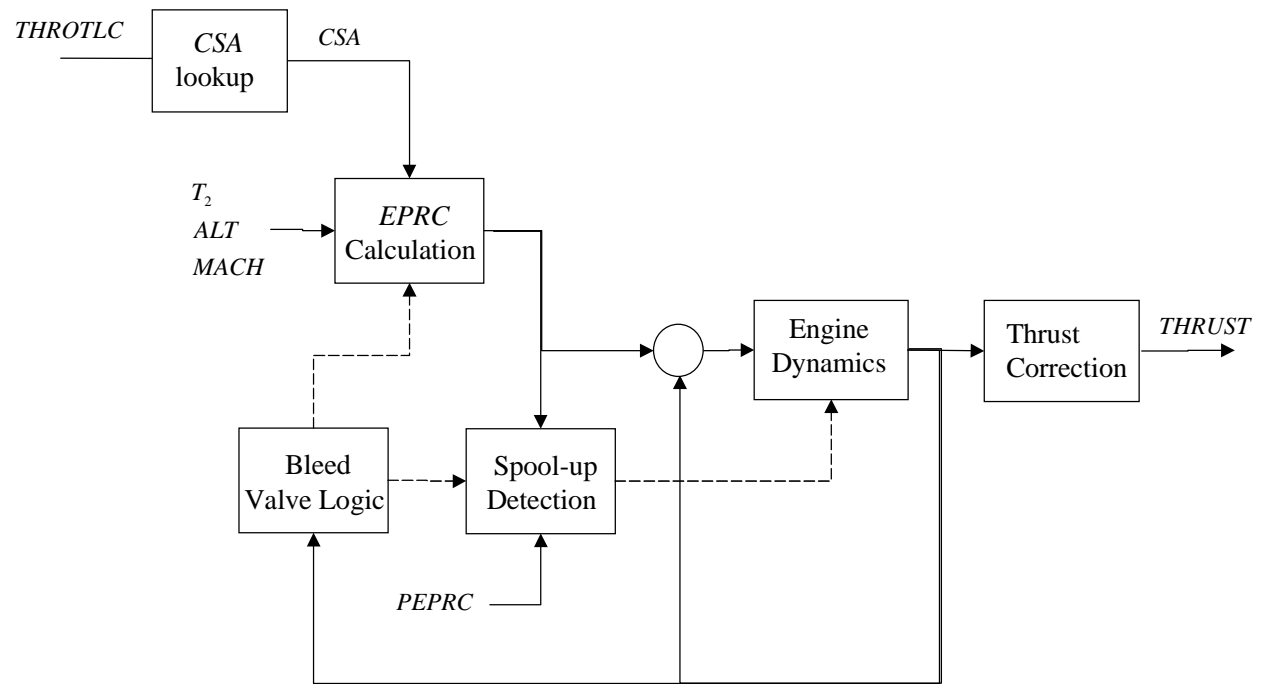


Figure 71. Engine model.



## Compressor Bleed Valve Operation

When the compressor blades turn too fast relative to the air moving through the engine, the effective angle of attack of each compressor blade becomes too great and can stall. Automatic bleed valves in the high-pressure section of the engine (station 3) open to dump air into the fan bypass air duct to increase the velocity of the air moving across the compressor to prevent this stalled condition. This condition can occur if the engine accelerates or decelerates too rapidly and the pressure drop across the compressor is too large. The difference in the static pressure sensed at station three and total pressure sensed at station two controls the operation of the compressor stall bleed valve. The critical EPR for this engine is between 1.110 and 1.128. A family of curves describes the relationship between the fuel control mechanism cross shaft angle (CSA) and the commanded EPR. The curves of commanded EPR as a function of CSA for various Mach numbers from 0 to .95 indicate the critical EPR transition by crossing at this EPR range [Ref. 9]. The bleed valve open logic chooses which curves to use to limit the rate of change of the EPR to account for the action of this valve system.

Compressor bleed air, compressed air from the rear of the compressor stage of the engine is used for deicing, turbine hot structures cooling, and aircraft heating. Engine bleed air can account for a power loss of 30 percent when all anti-icing systems are in use. This model does not account for the engine-deicing configuration.

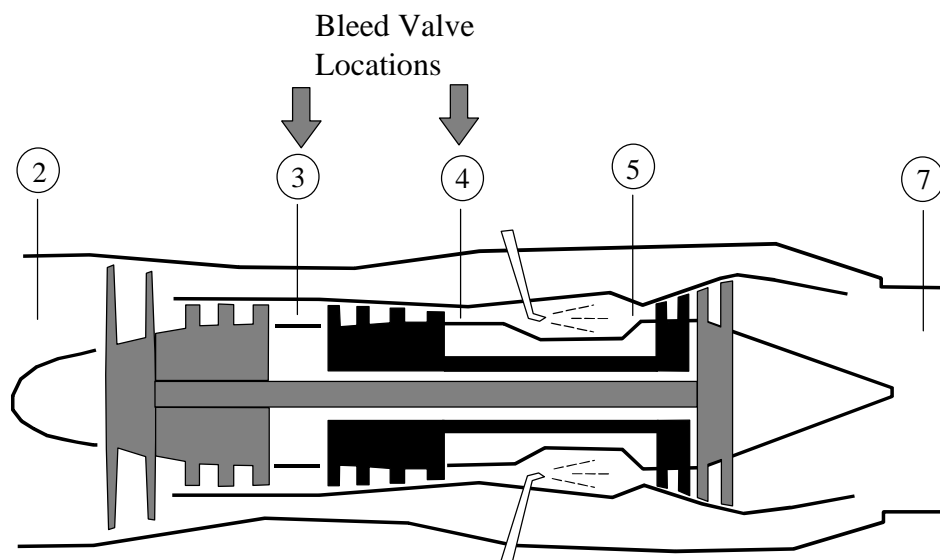


Figure 72. Engine compressor bleed air.



# Longitudinal Control Actuators and Servos

## Elevator

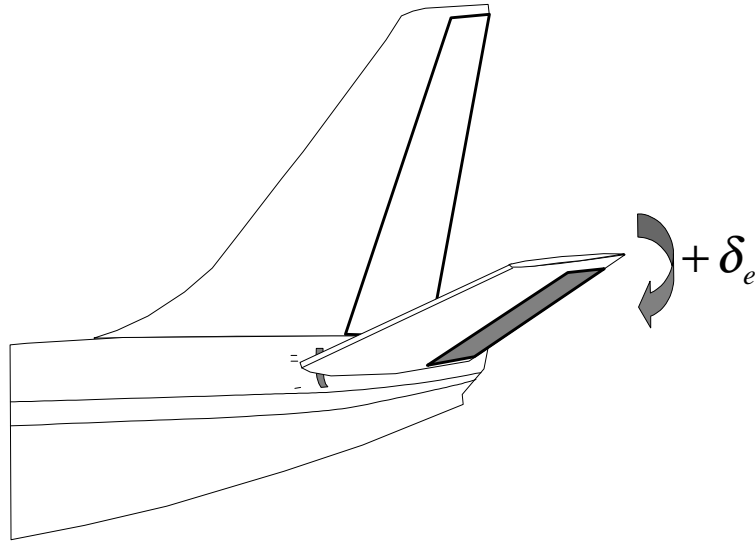


Figure 74. Elevator location.

The elevator is the primary pitch control input. It is used to set angle of attack and correct gust perturbations in the pitch axis. It is the trailing edge of the horizontal stabilizer. It is affected by the operation of the stabilator pitch trim system that rotates the entire horizontal stabilizer. Positive elevator deflection moves the elevator downward, which pitches the nose of the aircraft downward. The hinge moment is the torque produced at the elevator hinge by deflection of the elevator into the moving air stream. It is what must be overcome by the control system to set a given elevator deflection angle. Figure 76 is a plot of the hinge moment as a function of elevator deflection.



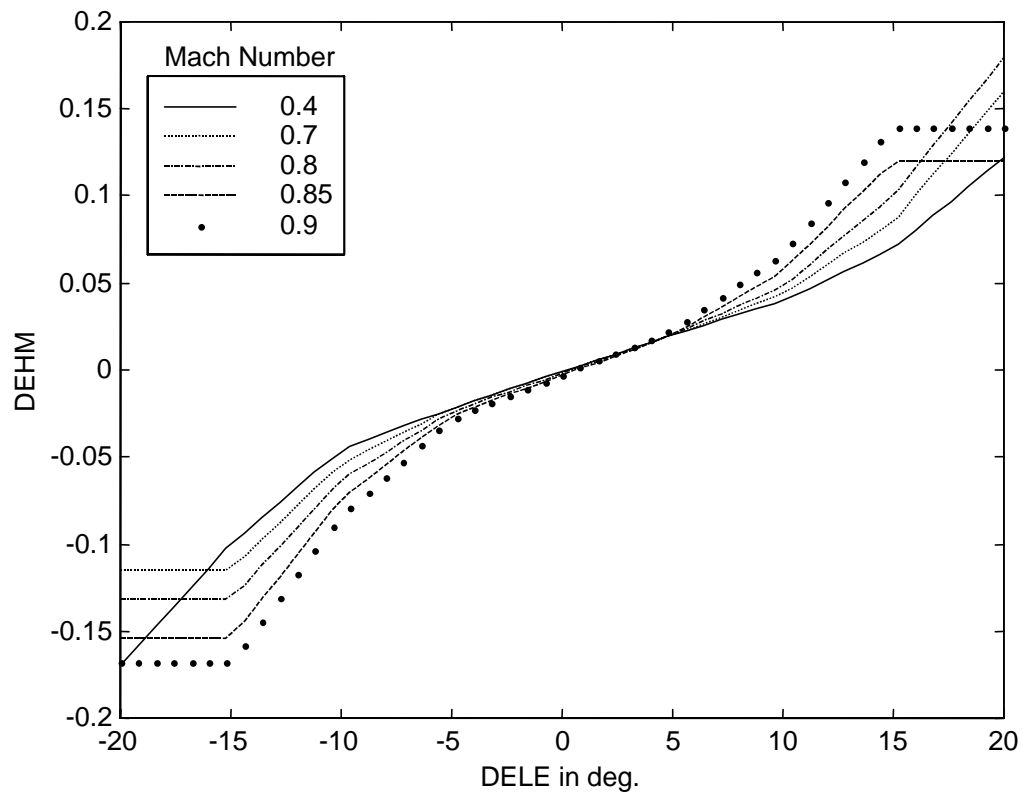
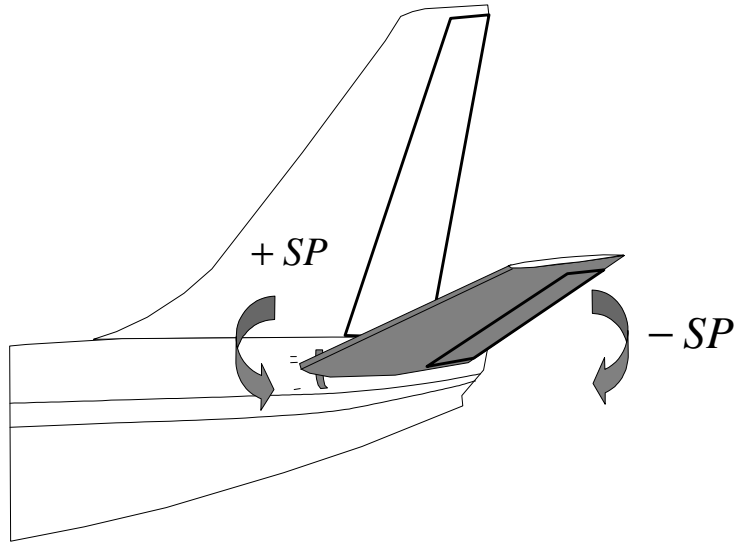


Figure 76. Hinge moment (DEHM) as a function of Mach number and elevator deflection (DELE).

## Stabilator



Stabilator (Horizontal Stabilizer and Elevator) Location

Figure 77. Stabilator location.

The Stabilator is the combined horizontal stabilizer and elevator. It moves slowly compared to the elevator and is used to make pitch trim adjustments. A positive movement of the stabilator moves the trailing edge up to pitch the aircraft up. It is operated by an automatic system that senses elevator deflection and attempts to reduce the elevator deflection once a deadband is exceeded.



# Lateral Control Actuators

## Aileron

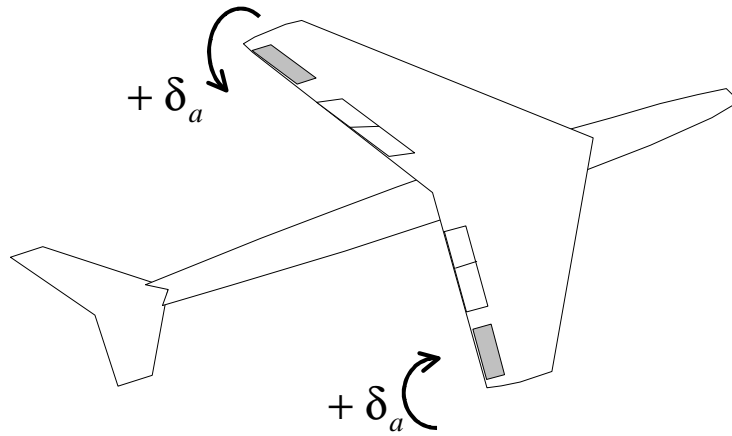


Figure 79. Aileron deflection angle.

The Aileron is used to yaw and roll the aircraft. It works with the spoiler system to accomplish this. A positive aileron deflection rolls the right wing downward. Yaw motion is coupled to this roll motion. A positive aileron deflection yaws the aircraft right.





## Spoiler

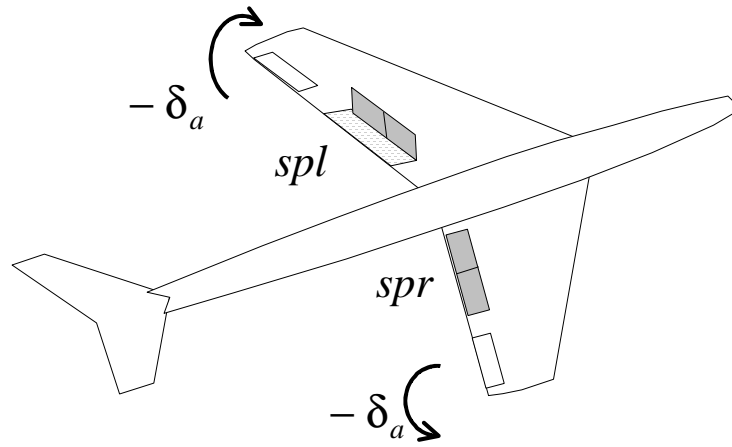


Figure 81. Spoilers.

The spoiler system is used to reduce lift and increase drag on the wing when it is deployed. It is used to assist the aileron system to roll and yaw the aircraft. The spoiler lifts on the downward rolling wing to balance the drag created by the downward deflected aileron on the opposite wing. This is to combat the tendency of aileron-adverse yaw caused by the increased drag of the downward aileron on the upward rolling wing. Spoilers only deploy once the aileron deflection exceeds a threshold. This is the “spoiler hysteresis”.

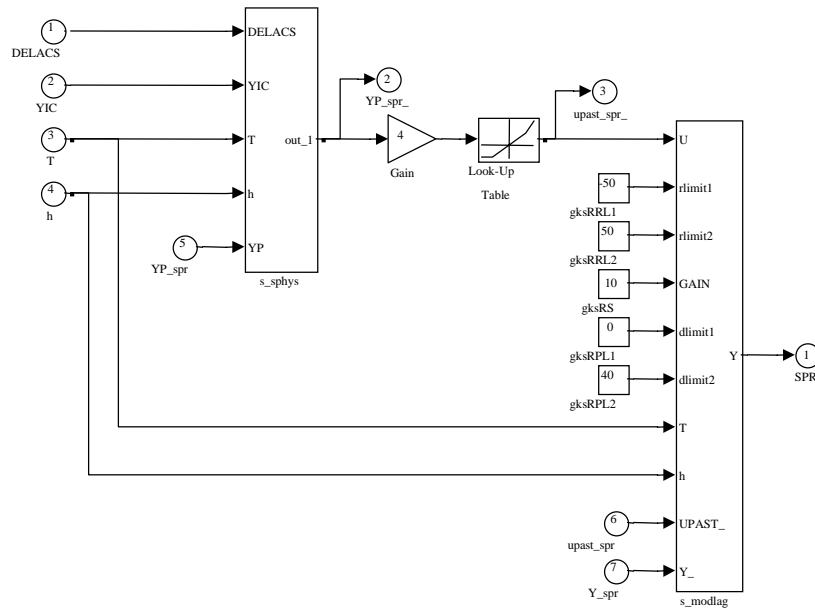


Figure 82. Spoiler servo Simulink diagram

## Rudder

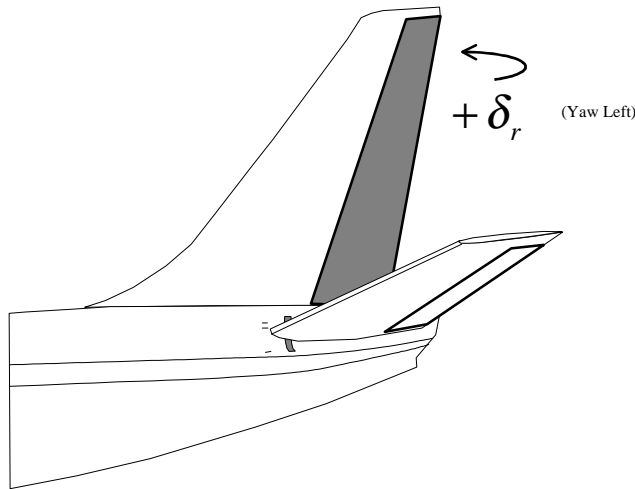


Figure 83. Rudder and yaw damper location

The rudder is used to control the yaw of the aircraft if the desired yaw is different from that given by the aileron and spoiler system for a given roll. It is mostly commanded automatically through the roll and yaw stability augmentation “yaw damper” system. The lateral control system only commands the rudder during the “decrab” maneuver during the latter phases of an automatic landing.

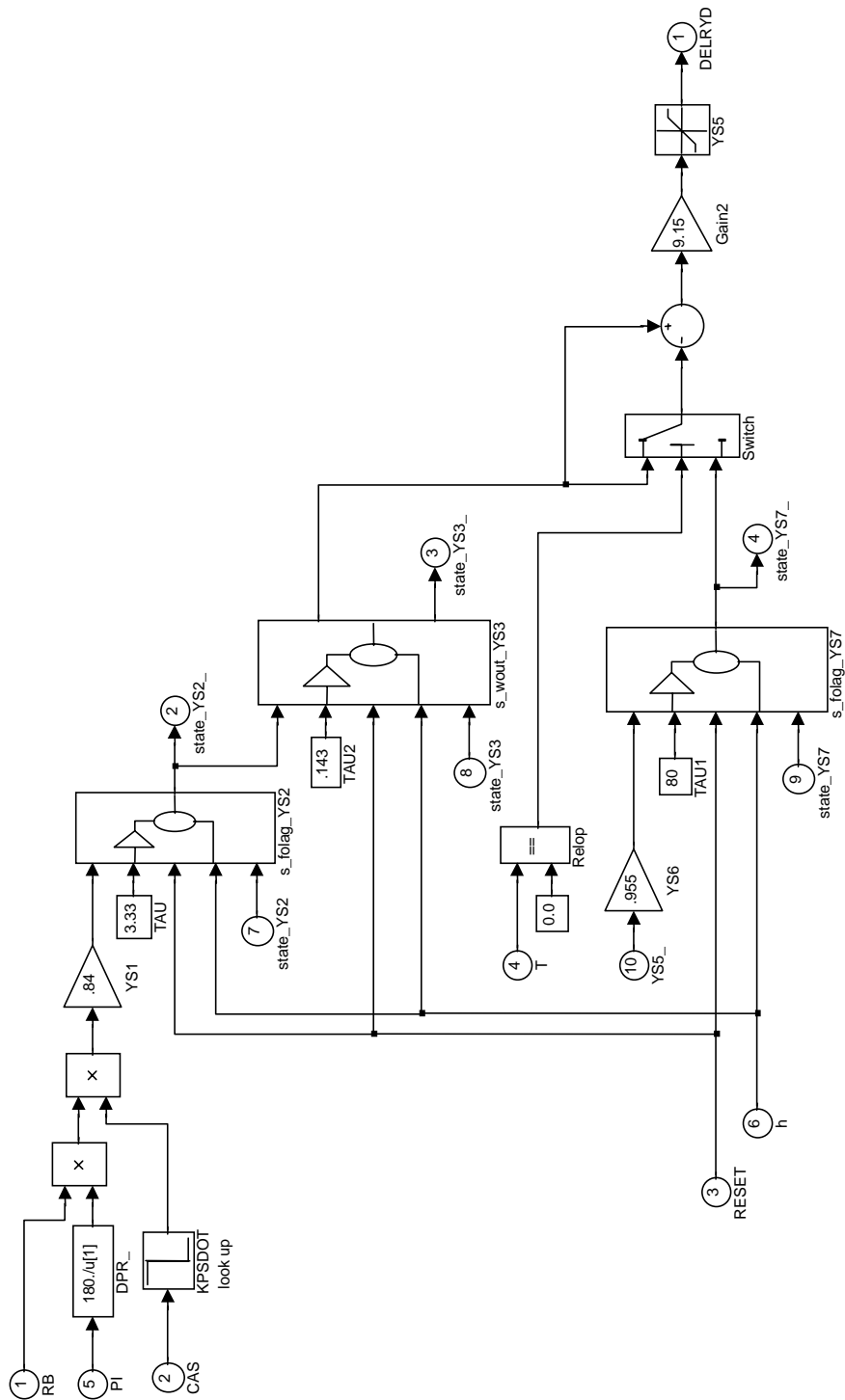


Figure 84. Yaw damper control system Simulink diagram.

## Approach to Conversion

The FORTRAN model was operated using Simulink to call the main functional blocks with FORTRAN Matlab executable (F-MEX) file blocks. This was done to get confidence with the Simulink method of simulation with the least change to the FORTRAN. Next the feedback loop was broken and the airframe state space model run with constant inputs to see if it would fly at all.

- Deterministic case, no winds and gusts
- Straight and Level Reference Condition
- State Space Equations and Integration and Rates Only
- FORTRAN S-function with Simulink Wrapper

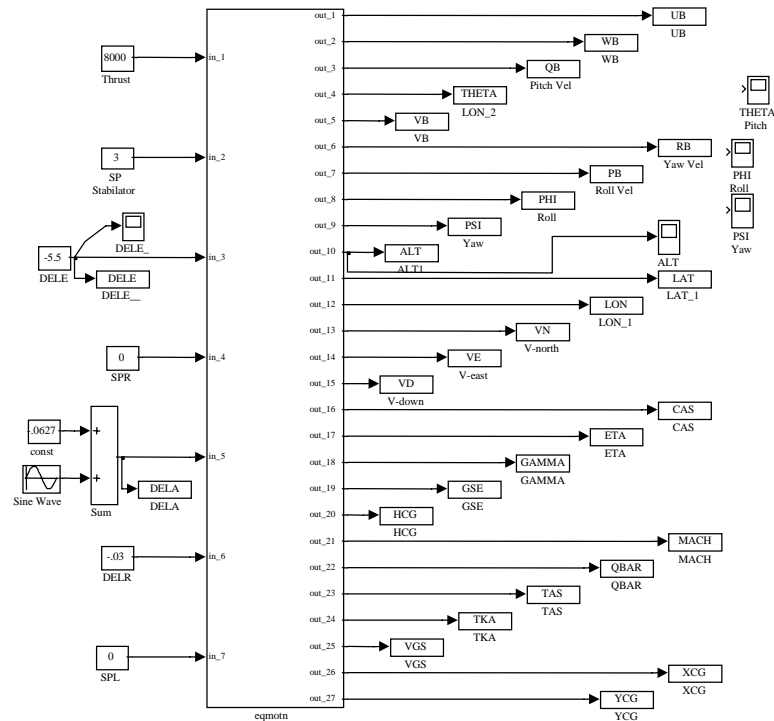


Figure 85. Simulink “wrapper” around FORTRAN deriv and eqmotn with fixed command inputs.

The open loop test was then made closer to the previous runs of the FORTRAN model by taking the time history of the control inputs as input to the airframe state space model without the autopilot present. This method of testing was repeated as parts of the airframe model were replaced with Simulink prototype modules.

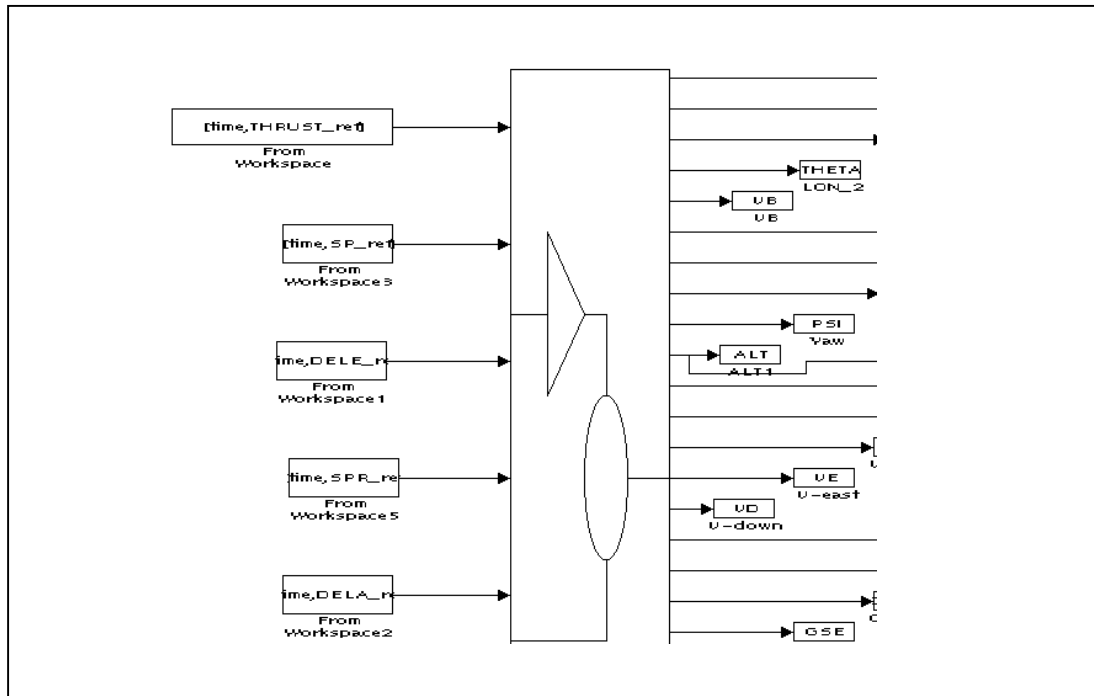


Figure 86. Simulate control input using FORTRAN simulation historical data.

The open-loop method of testing proceeded until autopilot modules were converted to Simulink. The large errors between the FORTRAN simulation output and the Simulink partial model were studied and worked on. A change would be tried to a Simulink module and its effect noted. If it had little effect and there was no engineering reason for it, it would be removed and another tried. This is the step-wise refinement technique. Sometimes conceptual errors were responsible and the aerodynamics text needed to be consulted. Another technique tried was use of frame by frame comparisons. The Matlab workspace output arrays made Matlab a convenient tool to use for this. The FORTRAN simulation and the Simulink simulation parameters were plotted on the same graph. Once they mostly agreed, the difference between them was plotted. The glitches were correlated with control surface movement and mode logic changes. Differences in time step values of parameters were seen and sequence and memory re-use problems were sought. More and more parameters needed to be plotted as the FORTRAN and Simulink output converged. It was necessary to carefully add additional engineering parameters to the FORTRAN simulation output to better study the parts of the Simulink simulation that had problems. The pitch rate parameter was put on a Simulink “scope” and plotted every time a test was done. Good and bad behavior characteristics became familiar and it was a good “barometer” of overall health of the simulation.

### ***Summary of FORTRAN to Simulink Porting Process***

- Use FORTRAN MEX Files for Missing Blocks
- Use Matlab Scripts for Nested IF Logic a good match for procedural language logic. Later found not to be compatible with code generation.
- Simulate Autopilot Control Inputs to the Aerodynamics Model Using FORTRAN Simulation Historical Data

- Open/Closed-Loop Testing
- Use Matlab Workspace to Plot, Plot, Plot...
- It was necessary to modify the FORTRAN simulation to plot additional parameters to better isolate errors.
- Macro error analysis
- Micro error analysis (step-by-step)
- The results of the previous FORTRAN simulation were repeated. It was learned that Matlab needed to be restarted if the test failed to have a valid subsequent test
- Attack easiest and largest errors first.
- Step-wise refinement
- Use Simulink “scope” to monitor QB. It was a good parameter for the pitch loop. Repeat it enough to get used to the pattern so any change is apparent.
- Open loop errors, investigate strongest divergence

#### **Problems Encountered**

- Transform to inertial space and divergence of body velocities UB1 and UB2
- Importance of time step alignment for open loop match
- Procedural vs. data flow languages and Integrators. The path not taken still is computed in the Simulink implementation unless explicit steps are taken to block it.
- Different values of pi used in FORTRAN model important for open loop match
- Constant parameters set in the Matlab workspace are not available to S-functions or code generation.

## Conclusion

The Airlabs FORTRAN B-737 Linear Autoland model was duplicated using Matlab/Simulink on several platforms (Unix, Windows 98 PC). Agreement of the model parameters was achieved to high precision (eight to nine significant figures). Since it is an extension of the B-737 model that was produced with company proprietary data removed, it can be used for research purposes. The Simulink block diagram language is a modern simulation tool and facilitates use by the control system research community.



## References

- [1] Barr, N.; Gangsaas, D.; and Schaeffer, D.: *Wind Models for Flight Simulator Certification of Landing and Approach Guidance and Control Systems.*, FAA-RD-74-206, December 1974.
- [2] *Cessna Manual of Flight*, Jeppesen Sanderson, Inc., 55 Inverness Drive East, Englewood, CO., 1996
- [3] Chalk, C.; Neal, T.; Harris, F.; and Pritchard, F.: *Background Information and User Guide for MIL-F-8785B(ASG), Military Specification – Flying Qualities of Piloted Airplanes.*, Technical Report AFFDL-TR-69-72, August, 1967.
- [4] DiStefano, J., III; Stubberud, A.; Williams, I.: *Schaum's Outline of Theory and Problems of Feedback and Control Systems*, McGraw-Hill Book Company, New York, 1967.
- [5] Eshbach, O.; and Souders, M.: *Handbook of Engineering Fundamentals*, Third Edition, John Wiley and Sons, Inc., New York, 1975.
- [6] Etkin, B.: *Dynamics of Flight*, John Wiley & Sons, New York, 1959.
- [7] Friedland, B.: *Control System Design, An Introduction to State Space Methods*, McGraw-Hill, Inc., New York, 1986.
- [8] Gainer, T.; and Hoffman, S.: *Summary of Transformation Equations and Equations of Motion Used in Free Flight and Wind-Tunnel Data Reduction And Analysis.*, Appendix A, page 67, NASA SP-3070, 1972.
- [9] Guenther, A.: *Linear 737 Autoland Simulation for Airlabs*, Sperry Corporation, Systems Management Group, Hampton, VA, Publication Number SP710-032, Contract NAS1-17929, 1985.
- [10] Gupta, S.; and Hasdorff, L.: *Fundamentals of Automatic Control*, John Wiley & Sons, Inc., New York, 1970
- [11] Halliday, D.; and Resnick, R.: *Fundamentals of Physics*, John Wiley & Sons, Inc., New York, 1970, p 197.
- [12] Klawans, B.: *Simple Method for Calculating the Characteristics of the Dutch Roll Motion of an Airplane.*, NACA-TN-3754, 1956.
- [13] Kerrebrock, J.: *Aircraft Engines and Gas Turbines*, MIT Press, Cambridge, Massachusetts, 1977
- [14] Kroes, M., Wild, T., Bent, R., and McKinley, J., *Aircraft Powerplants*, Sixth Edition, McGraw-Hill Inc., New York, 1990
- [15] Ogata, K.: *Discrete-Time Control Systems*, Prentice-Hall, Inc., New Jersey, 1987

- [16] Press, H.; and Meadows, M. T.: *A Re-evaluation of Gust-Load Statistics for Applications in Spectral Calculations.*, NACA-TN-3540, 1955.
- [17] Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B.: *Numerical Recipes in C The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge CB2 1RP, 1992.
- [18] Ragazzini, J.R.; and Bergen, A.R.: A Mathematical Technique for the Analysis of Linear Systems. *Proceedings of the IRE*, Vol 42, No 11, November 1954, pp. 1641-1657.
- [19] Treager, I. E.; *Aircraft Gas Turbine Engine Technology (Aviation Technology Series), Third Edition*, Glenco/McGraw-Hill, New York, 1996
- [20] Seckel, E., *Stability and Control of Airplanes and Helicopters*, Academic Press, New York, 1964
- [21] *U.S. Standard Atmosphere, 1962*, Washington 25, D.C., Bureau of Documents, Government Printing Office

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
01- 05 - 2004		Contractor Report			
4. TITLE AND SUBTITLE B-737 Linear Autoland Simulink Model			5a. CONTRACT NUMBER		
			NAS1-00135		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Hogge, Edward F.			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
			23-728-30-10		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Hampton, VA 23681-2199				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001				10. SPONSOR/MONITOR'S ACRONYM(S)  NASA	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)  NASA/CR-2004-213021	
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 08 Availability: NASA CASI (301) 621-0390      Distribution: Standard					
13. SUPPLEMENTARY NOTES Langley Technical Monitor: Celeste Belcastro An electronic version can be found at <a href="http://techreports.larc.nasa.gov/ltrs/">http://techreports.larc.nasa.gov/ltrs/</a> or <a href="http://ntrs.nasa.gov">http://ntrs.nasa.gov</a>					
14. ABSTRACT  The Linear Autoland Simulink model was created to be a modular test environment for testing of control system components in commercial aircraft. The input variables, physical laws, and referenced frames used are summarized. The state space theory underlying the model is surveyed and the location of the control actuators described. The equations used to realize the Dryden gust model to simulate winds and gusts are derived. A description of the pseudo-random number generation method used in the wind gust model is included. The longitudinal autopilot, lateral autopilot, automatic throttle autopilot, engine model and automatic trim devices are considered as subsystems. The experience in converting the Airlabs FORTRAN aircraft control system simulation to a graphical simulation tool (Matlab/Simulink) is described.					
15. SUBJECT TERMS 14. SUBJECT TERMS Aircraft simulation; Wind gust model; Control system; Autopilot; Computer model; State space model; Aircraft dynamics; Equations of motion; Commercial aircraft					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: <a href="mailto:help@sti.nasa.gov">help@sti.nasa.gov</a> )
U	U	U	UU	123	19b. TELEPHONE NUMBER (Include area code) (301) 621-0390